MIT Sloan School of Management

MIT Sloan School Working Paper 4714-08
9/8/2008

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

Jérémie Gallien and Théophane Weber

© 2008 Jérémie Gallien and Théophane Weber

All rights reserved. Short sections of text, not to exceed two paragraphs, may be quoted without explicit permission, provided that full credit including © notice is given to the source.

This paper also can be downloaded without charge from the
Social Science Research Network Electronic Paper Collection:
http://ssrn.com/abstract=1266171

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

Jérémie Gallien[2] and Théophane Weber[3]

September 8, 2008

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. In collaboration with a large US online retailer and using an extensive dataset of detailed flow information, we first develop a model with validated predictive accuracy for its warehouses operating under a waveless release policy. We then use that model to compute operational guidelines for dynamically controlling the main parameter of its waveless 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. Our 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 very serious consideration by practitioners. Facilities using a non-overlapping wave policy should also consider overlapping waves or a split sorter policy.

1 Introduction and Practice Review

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. A common warehouse design in such environments involves an automated sorter allowing different items in the same order to be picked upstream by different workers (Johnson 1998). While automated sorters take different forms, they all typically include an accumulation conveyor and a recirculating loop or "dogtrack" from which items are directed by a diversion mechanism towards individual chutes or lanes (also called "drop points") temporarily assigned to a single order or a small subset of orders with the same shipping destination. When all the relevant items have been diverted to a chute, these items can then be packed and/or loaded into a truck as a complete set. Disaggregating a possibly large set of orders into individual item picking instructions distributed simultane-

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

[2] 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

ously in several parts of the warehouse (*batch and zone picking*) and using such a sorter to subsequently re-aggregate these items downstream (since items from the same order may be picked by different workers in different zones) results then in average labor costs per order which can be significantly lower than with other system design alternatives (Choe, Sharp and Serfozo 1992, Petersen 2000, Bragg 2003).

The traditional approach for coordinating the flow of work through the warehouse in such systems 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, Sharp and Serfozo 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, Cook and Saipe 1979, Meller 1997). This approach presents several benefits:

- Using large wave sizes increases the density of items to be picked and thus picking labor productivity (Russell and Meller 2003, Le Duc and de Koster 2007), at least at the beginning and in the middle of the wave (see discussion below);
- Pick lists can be determined for all the workers simultaneously at specific points in time, and can be communicated using simple technology (i.e. paper printout). Indeed, the earliest descriptions of wave picking implementations in industry (e.g. Armstrong, Cook and Saipe 1979) predate by many years the advent of sophisticated computing and wireless communication technologies;
- 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 lanes (e.g. Armstrong, Cook and Saipe 1979, Johnson and Meller 2002, Owyong and Yih 2006).

However, many sources also discuss several important drawbacks associated with wave picking:

- Because the time for a picker to complete a wave is subject to both predictable and unpredictable variability, even when the work assignment within a wave is balanced across pickers and zones, some idle time may still occur. This may reduce the productivity of picking labor at the end of a wave and ultimately overall picking capacity (e.g. Choe, Sharp and Serfozo 1992, Petersen 2000, see also practitioners' discussions in Gilmore 2006a, Gilmore 2006b and Bradley 2007);

– While large wave sizes typically improve picking labor productivity, they also generate a large buffer of inventory (i.e. cycle stock) between picking and sorting. Such buffer is costly because of the corresponding accumulation conveyor equipment and floor space requirements (Russell and Meller 2003, Bradley 2007). Because of the relationship between wave sizes and number of sorter chutes pointed earlier, wave sizes also drive the sorter purchase cost, which can reach several millions of dollars for large facilities (Johnson and Lofgren 1994, Hinojosa 1996);

– As with any batch processing, waves add a cycle time component to order completion times, which can be an issue in environments with time-sensitive customers such as online retailing (Chew and Tang 1999, Johnson and Meller 2002, Le Duc and de Koster 2007);

– The sequential release into the sorter of non-overlapping waves with a number of orders roughly equal to the number of lanes may result in low capacity utilization. This is because the lanes corresponding to completed and/or packed orders cannot be re-assigned until the end of the current wave (Johnson and Lofgren 1994 and Johnson 1998). This issue is particularly critical during peak periods because sorters are very capital-intensive and thus often constitute the throughput bottleneck (Apple 2006, Perkins 2008, Gilmore 2006a and Bradley 2007);

– Finally, the workload of packers can be fairly concentrated at the end of each wave. This phenomenon can be modeled by using the realistic assumption that items are uniformly distributed within each wave (Hinojosa 1996, Johnson 1998) and considering for each order the expected maximum arrival time into a chute over all its items[4]; given automated sorter are typically not used for single item orders, most chutes only become ready to be packed in the last third of the wave (Hinojosa 1996, 2006), which results in relatively low packing labor productivity at the beginning.

As a result, more sophisticated forms of wave picking have been developed in order to mitigate these problems:

– To reduce pickers' idling at the end of waves, some companies allow different waves to overlap in the picking area, either across zones (Armstrong, Cook and Saipe 1979) or even within each zone (Owyong and Yih 2006). However, this practice creates the need for a pre-sorting operation downstream in order to separate items from different waves before

[4] The last item of an order with m items is then located on average at a relative position of $m/(m+1)$ within the wave.

release into the sorter and/or additional buffer space. In addition, differences of wave completion times across pickers and/or zones may accumulate over the course of a shift, unless an effective dynamic re-allocation of labor is performed;

– To increase sorter and packing labor utilization as well as overall throughput, 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 intermediate completion threshold such as 90% (Bozer, Quiroz and Sharp 1988) or 50% (Johnson and Lofgren 1994). The best threshold to employ is typically determined empirically or through simulation, and seems to vary widely across facilities. As pointed out in Johnson (1998), overlapping waves in the sorter also present some control challenges, in part because of the associated potential blocking effects when all the lanes and/or the recirculating buffer become full (see discussion below). A second strategy consists of splitting the sorter lanes in two, where each half is dedicated to a wave so that packers can work on a completed wave in one half while the next wave is being sorted into the other half (Ruben and Jacobs 1992, Russell and Meller 2003, Perkins 2008). In the ideal operating regime, the sorter completes the processing of that second wave shortly before packers complete the first one, so that packers are not starved for work during the transition between consecutive waves, and complete sorter lanes do not idle for long. Approaching this ideal requires a careful balance of wave completion times across packers, and between the packers and the sorter. To that end, some companies define several categories of workers according to their productivity and assign different numbers of orders or lanes to workers according to their category (with the fastest packers receiving the largest number of orders, see Ruben and Jacobs 1992). However, the pervasive unpredictable variability affecting the order packing and sorting times may easily disrupt this balance, thus reducing labor and/or equipment utilization. In addition, that strategy divides the largest possible wave size by two, which may impact picking labor productivity.

Because even these more sophisticated forms of wave picking still create challenges, a growing number of companies that include American Eagle Outfitters and Green Mountain Coffee have been using in their warehouses an alternative work 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 Forger 2005, Hinojosa 2006, Perry 2007, 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. As discussed earlier, 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 that is 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 picking loops with a fixed travel direction (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 the order corresponding to the relative positions of these items along the 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 by all pickers assigned to that zone, and re-assigns individual pickers across zones whenever imbalances between these completion times exceed specified thresholds. Finally, the buffer upstream of the sorter is used to delay within some limits the induction into the sorter of items belonging to orders with other items further upstream in the process (using for example one of the policies described in Johnson 1998), with the goal of reducing the accumulation time of orders once they are assigned to 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 active picker in the warehouse (typically provided by portable devices also including a bar-code reader and an LCD screen), and real-time centralized database management. The motivation for implementing this novel warehouse control policy however is that while it applies the principles of batch and zone picking and may thus achieve relatively high picking labor 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 because it does not increase towards the end of waves, 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 time before picking and the cycle stock 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 labor costs and throughput relative to traditional wave picking (Hinojosa 2006, Apple 2006, Cooke 2007, Perry 2007), and several support this claim with observations from actual implementations (Forger 2005, Bradley 2007, Morris 2008, McMahon 2008). Industry commentators have also described waveless picking as an application to warehouses of lean manufacturing principles, because of the lot size reduction it entails.

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. That 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 up. 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."

The work to be described here started from a collaboration with a leading US online retailer (*our industrial partner*), who had switched all of its warehouses with an automated sorter from wave-based to waveless picking before our interaction began. While our partner reports observing then significant increases in throughput, equipment utilization and labor productivity, it did not initially impose formal guidelines for how managers should dynamically adjust the primary control levers of this new order release policy, and indeed experienced gridlock more frequently than desired during the peak demand seasons following that implementation. This situation motivated in part the two research objectives pursued in this paper, using an extensive dataset of detailed flow information obtained from our industrial partner: (Objective 1) *Develop a quantitative model to generate operational control guidelines for waveless picking during peak demand periods, with the goal of maximizing throughput while keeping the likelihood of gridlock sufficiently low*; and (Objective 2) *Leverage this model to conduct a rigorous performance comparison between wave-based and waveless release policies in the context of our industrial partner's warehouses.* After a discussion of the related literature and our contributions in §2, we describe our work on the first objective stated above in §3. We then present a quantitative model describing wave-based picking in the context of our partner's warehouses in §4, and discuss in §5 the simulation experiments we performed in order to achieve our second objective. Concluding remarks are provided in §6, 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 Related Literature and Paper Contributions

We limit our discussion to papers specifically motivated by warehouses with an automated sorter, and we refer the reader to the surveys by de Koster, Le-Duc and Jan Roodbergen (2007) and Gu, Goetschalckx and McGinnis (2007) for recent and extensive reviews of the many design and control problems arising with other types of warehouses. Also, literature that is methodologically related to our analysis is discussed in subsequent sections.

A first set of papers positively answers the question of whether an automated sorter constitutes a justified design choice in some environments. This question is closely related to the choice of an order picking policy, because the need for sortation only arises when pick lists do not preserve order integrity. Using simple queueing models, Choe, Sharp and Serfozo (1992) compare the order cycle times associated with the three strategies of single order picking, batch picking and batch zone (wave) picking with an automated sorter. With more complex simulation models, Petersen (2000) investigates the policy of sequential zone picking in addition to the three others just mentioned, considering not just order cycle time but also labor requirements. Finally, Russell and Meller (2003) develop a simple deterministic cost model to inform the decision of whether manual or automated sorting should be used in a warehouse operating under wave picking. In the specific environment of online retailer Amazon.com, Bragg (2003) also considers similar simple deterministic cost model to assess various warehouse design alternatives.

A second group describes the use of simulation models to determine various dimensioning and control parameters of warehouses with automated sorters, given specific design objectives. Bozer and Sharp (1985) explore the impact on sorter throughput of the number of sorter lanes and their storage capacity as well as the use of recirculation and the concentration of items from the same order within a wave. That study is complemented by Bozer, Quiroz and Sharp (1988) who also investigate the throughput implications of the wave profile (size, distribution of items per order), the lane assignment policy and the degree to which consecutive waves are allowed to overlap. Finally, Johnson and Lofgren (1994) report the successful use of simulation model decomposition when designing a new warehouse with an automated sorter. This decomposition is enabled by sufficient picking capacity together with the wave release strategy, which effectively decouples the areas of picking and sorting. Also relevant to sorter design decisions is Johnson and Meller (2002), which presents an analytical model predicting the throughput of induction stations used in split-case sorting operations, where workers need to manually place incoming items onto moving trays bound to sorter chutes.

A last set explores more specific operational problems motivated by the use of automated sorters. We first note that, although not reviewed here, some of the papers investigating the problems of storage assignment, zoning, order batching and picker routing within the order picking area considered in isolation (see de Koster, Le-Duc and Jan Roodbergen 2007) are also potentially relevant to the operations we focus on. However, only a couple develop models that capture the impact of these decisions on the sorting process. Among them, Armstrong, Cook and Saipe (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. In the context of module-based fulfilment centers, Owyong and Yih (2006) present a heuristic for modifying pick lists to reduce order consolidation time or *chute-dwell time*, which is the time between the arrivals of the first and last item of a customer order in a sorter chute. Finally, Meller (1997) describes integer programming models for the problem of assigning orders to sorter lanes (lane assignment), assuming that the sequence of incoming items is known. Relaxing this last assumption, Johnson (1998) develops an analytical model predicting the impact of various lane assignment strategies on expected total wave sorting time.

In all the literature just discussed, wave picking is the only operating policy considered for warehouses with an automated sorter. To the best of our knowledge, the present paper is the first to present a mathematical model for the problem of order release control in a warehouse operating under waveless picking (see §1 for background), and to address quantitatively the questions of whether and how gridlock may be avoided under such policy. This study also leads us to investigate and characterize the relationship between the release strategy for picking orders and the sorter operation, which has been consistently described as an important object of research (Johnson 1998, Petersen 2000, Owyong and Yih 2006). Finally, the solution we develop for the order release problem mentioned above enables us to present the first meaningful quantitative performance comparison between wave-based and waveless picking in a specific context.

3 Waveless Release Model and Analysis

As part of the waveless picking policy described in §1, the main daily flow control levers available to our partner 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, which can be adjusted a few times per day (we refer the interested reader to the Online Appendix for a detailed description of our partner's warehouses). Note that staffing decisions for induction stations (and more generally their capacity) are not explicitly considered in this section. This is because our partner uses induction stations with automated coordinated induction belts which were 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. Also, the differences in possible control change frequencies just stated justify our focus on the revolving batch size under the assumption that staffing levels constitute fixed exogenous parameters.

The revolving batch size affects the overall picking rate in two ways. First and foremost, through the resulting density of items to be picked along the picking paths. Secondly, through the starvation of pickers occuring when the revolving batch size is set to a low value[5]. Fortunately, our partner had developed through extensive empirical studies a simple but reliable model that determines the size of the revolving batch required in its environment in order to generate a specified average overall picking rate under a given staffing level for pickers. However, no formal guidelines were available at the outset of our interaction for dynamically changing the target picking rate as a function of observed process conditions (conveyor system congestion, numbers of complete, incomplete and unassigned sorter chutes) and packers' staffing level[6]. This was mostly problematic during peak demand period, when the pressures for high throughput sometimes resulted in gridlock (see §1).

In the remainder of this section, we present a quantitative model and analysis developed to identify flow control strategies for waveless picking that maximize throughput while keeping the probability of gridlock under an acceptably low level during such peak demand periods. We define our predictive model in §3.1, present related approximate dynamics and discuss their validation in §3.2, state the optimization problem we consider in §3.3 and finally discuss an associated solution algorithm in §3.4.

3.1 Predictive Model Our model for predicting congestion in our partner's warehouse as a function of the flow control policy employed is a serial queueing network with three stations and features similar to that found in Gallien and Wein (2001). It is defined as follows:

[5] 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.

[6] Instead, managers tended to apply 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 any analysis, appeared inconsistent across managers and facilities, and adherence to those guidelines was not enforced.

– Each circulating entity in this network represents a customer order, and the arrival rate of these orders to the network over time is the primary control we ultimately seek to optimize (in §3.3). This input is modeled as a Poisson process whose rate at time $\tau \geq 0$ is denoted $\lambda(\tau)$, and the associated release rate control policy will be noted $\boldsymbol{\lambda}$. In the real system, $\lambda(\tau)$ corresponds to the average current rate at which orders are picked. As in Russell and Meller (2003), we thus do not explicitly model the warehouse layout, stowing policies and picker routing policies employed; however we do consider their aggregate impact on the overall rate at which pickers release items onto the conveyor system. Specifically, this impact is captured by the empirical model mentioned above, which links overall average picking rate with the revolving batch size and the pickers' staffing level, and thus enables a practical implementation of flow control policies characterized by target order picking rates (converting average order pick rate to average item pick rate is straightforward since the average number of items per order $\mathbb{E}[M]$ is easily determined). This modular approach is more tractable, but also alleviates the need to explicitly model features which tend to be more idiosynchratic such as the layout of the picking area. In other settings, the relationship between batch size, staffing levels and average overall picking rate may also be determined empirically, or through analytical models of the kind developed by Chew and Tang (1999) and Le-Duc and de Koster (2007). We also 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 analytical tractability considerations and the intuitive relevance of Palm's theorem to this setting. Because the maximum picking rate achievable is limited in practice by various factors, we assume an upper bound $\bar{\lambda}$ for this control. During the peak demand periods that we are most concerned with, the virtual queue of orders placed by customers but not yet released for picking is always sizeable and never empties. As a result, our assumption that the upper bound $\bar{\lambda}$ is exogenous seems reasonable. Finally, because of practical database synchronization issues, the frequency at which the revolving batch size may be adjusted is limited. As a result, we assume that the average picking rate $\lambda(\tau)$ may only be changed in our model at discrete time points separated by a period δ (of the order of a few minutes). As a result, the release policies $\boldsymbol{\lambda}$ considered effectively implement 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 station of this queueing model has an infinite number of servers, each with identically distributed service times following a distribution noted A and representing the *time-to-chute*, or time between the first time at which an item belonging to a customer order is released for picking anywhere in the warehouse and the first time at which an 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 a finite number of servers equal to the number of sorter chutes n, each with identically distributed service times following a distribution noted B and representing the *chute-dwell time* of every order, which is the time that each customer order spends in a chute before all its items are complete. The number of orders undergoing service in this second station thus represents the number of incomplete chutes in the sorter at any point in time, and follows a process denoted $Y(\tau)$. As before, we define $Y_t \triangleq Y(t\delta)$.
- Finally, 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. Its values at the discrete time points $(t\delta)_{t\in\mathbb{N}}$ are also denoted $Z_t \triangleq Z(t\delta)$.

The considerations that led us to formulate the predictive model just described are the following:

- The primary model data $(A, B, C, \bar{\lambda}, w)$ is readily available in practice. This is because our partner's warehouses use a sophisticated data collection system involving bar-code scanners carried by pickers and packers and also placed in many locations in the conveyor system, induction stations and sorter chutes. This system generates a database of detailed flow timing information for individual orders from which our model's service time distributions can be constructed;

- The primary model output $(X(\tau), Y(\tau), Z(\tau))$ is directly observable in practice, and in fact that information corresponds exactly to that actually observed by managers when adjusting the revolving batch size. This model feature thus guarantees that the informational requirements of any closed loop release control policy developed will be realistic, and also allows for a quantitative model validation to be performed (see §3.2.2);
- Note that $Y(\tau) + Z(\tau)$ represents at any time τ the total number of busy chutes (either incomplete, complete and waiting for a packer, or being packed). The occurence of gridlock can thus be directly expressed in terms of our model's primary output as the event $Y(\tau) + Z(\tau) > n$, where n denotes the total number of sorter chutes. Besides, consider any release control policy $\boldsymbol{\lambda}$ ensuring with sufficiently high probability that this event does not occur (we discuss in §3.3 how this may be achieved). Under such policy the seemingly salient model assumption of infinite buffer size at the third station is in fact immaterial.

The appealing features of this model formulation do come with a price however. The actual time-to-chute and chute-dwell time of any particular order depend directly on the transit times between the picking area and the sorter of all the items it contains. In turn, those transit times are affected in practice by the congestion upstream of the sorter[7]. But the congestion upstream of the sorter is directly related to the output process $X(\tau)$ representing the number of orders in the first station of our model. In summary, the service times A and B of the first two stations in our model could not a priori be considered exogenous, let alone stationary[8]. While one could conceivably attempt to create an analytical model predicting these service times as a function of the release rate and/or output process $\{(X(\tau), Y(\tau), Z(\tau)) : \tau \geq 0\}$, we believe that such model would be considerably more complicated than ours. 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 more precisely as the total number of items $I(\tau)$ on the conveyor

[7] The Online Appendix contains a mathematical statement of the relationship between transit times and time-to-chute and chute-dwell time, as well as a study of the dependence of transit times on congestion.

[8] The same observation could conceivably be made for the service times of the third station because the walking time of packers also appears endogenous. However, data shows that C is in fact fairly stationary.

system between the picking area and the sorter. We then fit the distributions of A and B for all time periods in our data set when the system was in each congestion level, and thus obtain empirical distributions $A(g)$ and $B(g)$ for all the congestion levels g just defined. Remarkably, the distributions $A(g)$ and $B(g)$ constructed by this method are unimodal for each g (whereas the empirical distribution of A and B obtained for the entire data set are not), thus validating our approach at a qualitative level (see the Online Appendix for empirical density plots of these service time distributions).

Finally, the last modeling step consists of identifying a relationship between the total number of items $I(\tau)$ on the conveyor system (which characterizes the congestion level g) and the output $(X(\tau), Y(\tau), Z(\tau))$, so that the dynamics of our model be well defined. With $\mathbb{E}[M]$ denoting the average number of items per customer order as before, the expression $\mathbb{E}[M]X(\tau)$ results in a significant underestimation of the number of items $I(\tau)$ in process between the picking area and the sorter, because many items still on the conveyors belong to customer orders with one or several other items already in a chute, and are therefore not accounted for by the process X. However, the expression $I(\tau) \approx \mathbb{E}[M](X(\tau) + Y(\tau)/2)$ provides a relatively accurate estimate for the total number of items on the conveyors – this expression corresponds to the approximation that the order statistics of the arrival times of the items of each shipment to their corresponding chute are equally spaced in expectation. Ultimately, this method and the resulting predictive accuracy of our model are validated by performing a comparison of actual system output and simulated model output over time for given starting conditions and picking rate history (see §3.2.2).

3.2 Approximate Dynamics

3.2.1 Derivation Our next step is to study the dynamics of the queueing model described in §3.1, that is characterize how the process (X_t, Y_t, Z_t) evolves over time as a function of any release control policy $\boldsymbol{\lambda}$ considered. We want to understand in particular how such policy may dynamically affect the likelihood of the gridlock event $\{Y_t + Z_t > n\}$. Unfortunately, the exact analysis of that queueing model appears challenging because (i) the short control time period δ precludes the use of any steady-state analysis; and (ii) service time distributions for the first two stations depend on the congestion level and are thus state-dependent. These observations motivate the development of an approximate version of our queueing model that is more amenable to analysis but still offer a suitably realistic representation of the actual pick-to-ship process described earlier. We specifically consider the following approximations:

- *The second queue has an infinite number of servers.* This assumption is justified in light of the optimization problem that we consider eventually (in §3.3), where the probability of the gridlock event $\{Y(\tau)+Z(\tau) > n\}$, which contains the event $\{Y(\tau) > n)$, is constrained to be very small.

- *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 have coefficients of variation that are close to 1 and shapes that are similar to that of an exponential (see Online Appendix). However, the empirical distributions we constructed for $A(g)$, which have positive and relatively large support lower bounds determined by the speed of conveyors, do not. Note that Weber (2005) derives more accurate system dynamics for this model through an approximating queueing network that is still Markovian, using phase-type distributions and results on the $M_t/G/\infty$ queue from Eick, Massey and Whitt (1993). However, we have found that the resulting increase of the number of state space dimensions does increase computational requirements for our approximate DP algorithm to a level that is impractical, at least at the time of writing.

- *Orders move at most one station downstream during each time period $[t\delta;(t+1)\delta)$.* This approximation substantially simplifies system dynamics, and does not seem to much harm prediction accuracy: 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 $[t\delta;(t+1)\delta)$.* When simulating the exact queueing dynamics for the model described in §3.1 under various policies and input parameters, we have found that consecutive changes of congestion level occuring less than δ time units apart were very rare.

- *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 seemingly well in simulations resulted 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 related derivations of transient system dynamics)[9]:

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

where the four random variables $(N_t^{\to X}, N_t^{X\to Y}, N_t^{Y\to Z}, N_t^{Z\to})$ 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. Computations can be even further reduced by substituting the binomial variables $N_t^{X\to Y}$ and $N_t^{Y\to Z}$ 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.

3.2.2 Validation Our next step was to validate the approximate queueing dynamics (1) using numerical simulation; the high-level procedure was 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 collected data series recording the actual state evolution $(x^*(\tau), y^*(\tau), z^*(\tau))$, actual control history $\lambda^*(\tau)$ and actual number of staffed packers $w^*(\tau)$, each with one data point per minute and spanning a period of several days during a peak demand period faced by our industrial partner. In the context of the order release control problem that we formally define in the next section, a particularly relevant prediction lead-time is the control period δ (of the order of a few minutes), since the associated dynamic program involves an expectation of the value function at time $\tau + \delta$ given the system state at time τ. 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

[9] 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. This assumption is appropriate given the very high utilization of that queue for most of the scenarios considered (particularly when nearing the gridlock state), and in the other scenarios we verified that the associated expression was still an acceptable approximation.

simulated the random variables $(X_{t+1}, Y_{t+1}, Z_{t+1})$ characterized by (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 (say) the estimated mean 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, due to space constraints we only report here actual and predicted values for the number of busy chutes, which is particularly relevant in this context because the occurence of gridlock is modeled by the event $\{Y_t + Z_t > n\}$. Specifically, Figure 1 shows the mean $\mathrm{E}[Y_{t+1} + Z_{t+1}]$ and associated centered empirical range with length $6\sigma[Y_{t+1} + Z_{t+1}]$ thus estimated at each record time point (τ) over one full (representative) day, along with the actual corresponding 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 overestimation follows from the fact that our queueing model does not capture explicitly the induction stations at which items coming from the conveyor system are individually placed on the sorter's tilting trays. Indeed, the transition rate between the first and second stations in our queueing model only depends on the number of orders in the first station as well as its service time (time-to-chute) distribution, and thus does not directly account for the staffing of induction stations. This modeling choice is justified during the regular (non-highlighted) working hours, as the induction stations have appropriate processing capacity then. However, the periods highlighted in Figure 1 drastically impact the staffing of these induction stations, so that the actual flow of items into the sorter then is either considerably reduced (maintenance/breakdown) or stopped

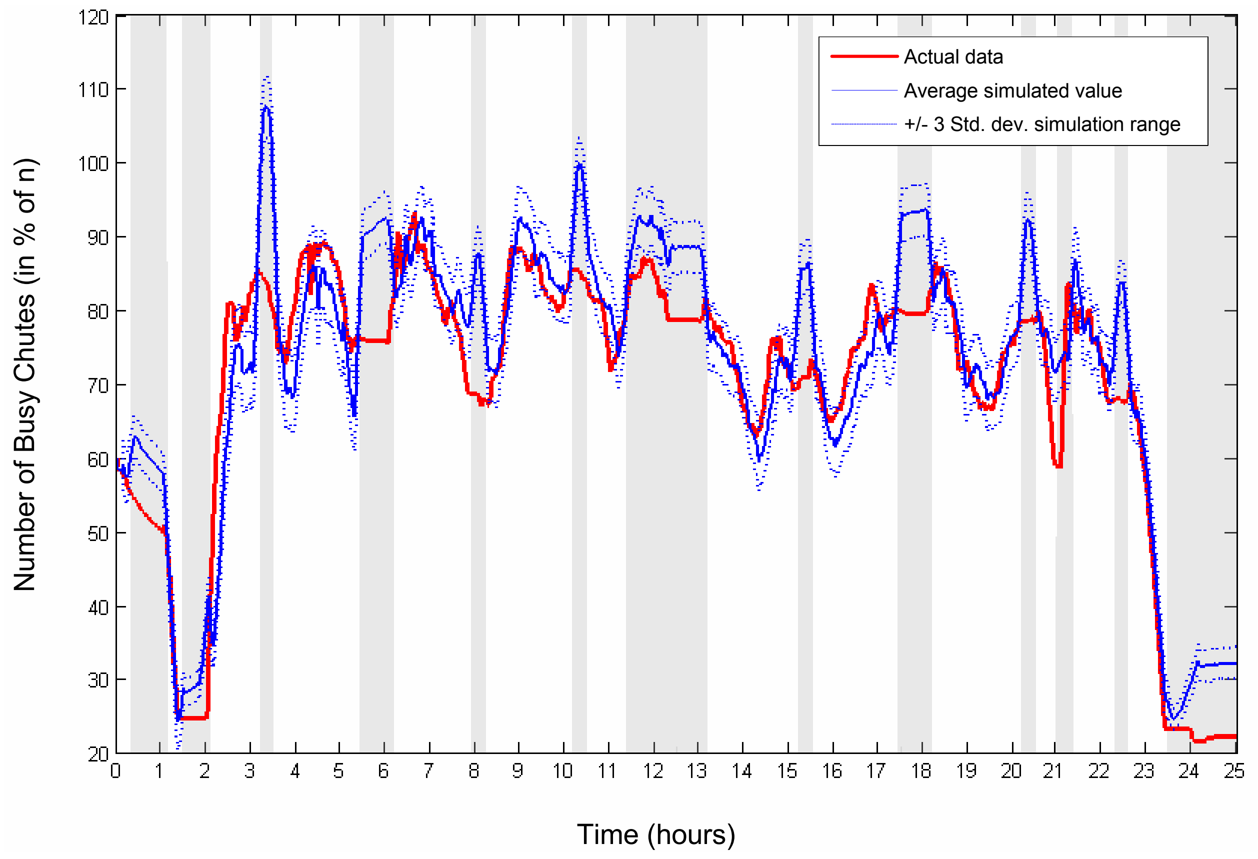


Figure 1: Predicted Distribution $Y_{t+1} + Z_{t+1}$ and Actual Value $y^*(\tau + \delta) + z^*(\tau + \delta)$ of the Number of Busy Chutes over a 24 Hour Period.

(work breaks). Indeed, note that the actual number of busy chutes remains constant during all work breaks listed above, which reflects that actual flows into the sorter (induction) and out of it (packing) are stopped then. While the model correctly captures the packing rate reduction during such periods through its input data w, it ignores the corresponding decrease in induction rate, leading to the overestimation observed.

While the results observed during the highlighted periods enhanced our understanding of the relationship between our model and the actual system, they did not seem relevant to validation since gridlock may not occur during periods of forced reduced activity. During regular working hours the actual number of busy chutes observed almost always lied within our model's predicted range, so that the approximate dynamics tested appeared sufficiently accurate given our purposes; this validation exercise was thus deemed conclusive.

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

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

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 from the present context however, we will almost always omit that dependence in the following.

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 a Bernoulli probability that the future stream of rewards may be interrupted 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 §3.1. In addition, we may consider α for practical purposes as a tuning parameter affecting the features and performance of the policies derived. 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[10], which are proven in Blackwell (1962) and hold for any initial state:

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

[10] Blackwell proves the first equality in each line for a finite state Markov Chain. The second equality in the second line follows from the ergodicity of Markov chains with unique steady-state distributions.

The second and third equality statements in (3) imply that the first constraint in (2) is asymptotically equivalent as $\alpha \to 1$ (a very relevant regime given the numerical value we use for α are very close to 1) to the much 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 comparably 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 §3.4). From a modeling perspective, that constraint thus balances the throughput maximization objective in (2) with the need to avoid the state of 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 resulting in some 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 the discount factor α that are 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.

3.4 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 can be easily derived (see Altman 1999 for a review, and Hordijk and Spieksma 1989 for an application of these methods to a queueing model with features similar to ours). In particular, we now outline a method for computing a solution to $CDP[\beta]$ by solving a sequence of related unconstrained dynamic programs $UDP[\theta]$ 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 $CDP[\beta]$. The problem $UDP[\theta]$ just defined is thus a Lagrangian relaxation of $CDP[\beta]$ where the first constraint in (2) is now captured through the objective function and weighted there by the multiplier θ, to be interpreted as an instant penalty for entering a gridlock state, i.e. $\{Y_t + Z_t > n\}$. 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 $\mathbf{c}^\theta(x, y, z) \triangleq \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\{Y_t^{\boldsymbol{\lambda}^\theta} + Z_t^{\boldsymbol{\lambda}^\theta} > n\}} | (X_0, Y_0, Z_0) = (x, y, z)]$, where $\boldsymbol{\lambda}^\theta$ is an optimal policy for $UDP[\theta]$. The following results are obtained through a straightforward adaptation 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):

Lemma 1 *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 $UDP[\theta^*]$.*

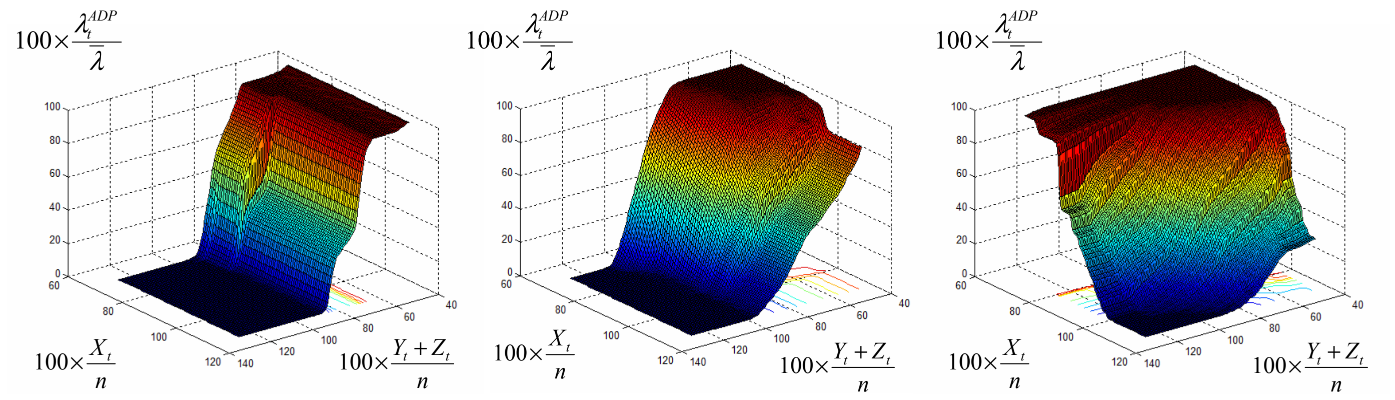
Lemma 2 *Suppose that for some $\theta \geq 0$ there exists a policy $\boldsymbol{\lambda}^\theta$ such that $\boldsymbol{\lambda}^\theta$ is optimal for $UDP[\theta]$ and achieves $\mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\{Y_t + Z_t > n\}} | (X_0, Y_0, Z_0) = (x, y, z)] = \beta$. Then $\boldsymbol{\lambda}^\theta$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

The solution method we have implemented consists of 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 (see below), 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 $[\underline{\theta}, \bar{\theta}]$, with an accuracy termination parameter ϵ. While a more detailed description and discussion of convergence properties can be found in the Online Appendix, we observe here that this algorithm may require to solve up to $\log_2(\frac{\bar{\theta} - \underline{\theta}}{\epsilon})$ unconstrained dynamic programs $UDP[\theta]$ in order to compute a solution to $CDP[\beta]$. The associated computational efforts may thus seem daunting at first. However, we were able to reduce computations to a practical level through the following additional steps: (i) In order to solve each instance of $UDP[\theta]$, we use an approximate policy iteration algorithm relying on the Robbins-Monro stochastic approximation scheme for the evaluation step, and Monte-Carlo simulations for the improvement step (see the Appendix for a detailed statement of this algorithm, related references and discussion); and (ii) At the k-th iteration of the search algorithm (with corresponding multiplier value θ^k), we use the best policy found for $UDP[\theta^{k-1}]$ at the previous iteration as a starting point to the policy iteration algorithm used to solve problem $UDP[\theta^k]$.

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 $\lambda_t^{ADP} \triangleq \lambda^{ADP}(X_t, Y_t, Z_t)$.

3.5 Policy Structure We now discuss the qualitative features of policy ADP^β. Theoretical results on the structure of optimal policies for problem (2) have so far eluded us, which is relatively unsurprising given the relative complexity of the underlying model (a queueing network with discrete-time controls and service times depending on a function of congestion in some parts of the network)[11]. As a result, the following discussion is based instead on a large number of consistent empirical observations.



(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 $\beta(1-\alpha) = 10^{-6}$.

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

[11] While it is conceivable that such results could be obtained assuming simplified model dynamics (with exogenous service times for example), such a study falls outside the scope of the present paper, in part because validation experiments of the type described in §3.2.2 show that simplified versions of the model we consider have poor predictive accuracy in the industrial setting motivating our study.

$(X_t/n, (Y_t + Z_t)/n)$[12] 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 intuitive feature observed 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 (this region of the state space corresponds to the upper 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 §3.2.1). Indeed, simulation shows that these drops enable the policy to maintain the system in a desirable steady-state congestion level[13].

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 must heavily utilize packing capacity, and any given change in the arrival rate of orders to the third packing station has a substantial impact on its occupancy process, or number of green chutes Z_t. Consequently, the ADP^{β} policy thus 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 relevant regions of the state space. 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 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

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

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

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 tend to be attended to immediately by a packer, i.e. the jobs representing them in our model experience little or no queueing in the third station. 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 §3.1).

4 Wave Release Models

The model described in the previous section was primarily constructed with the goal of helping our industrial partner optimize its waveless release policy. The present section discusses how it can also be modified in order to support a partial performance comparison with wave-based policies through simulation. 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 (the primary concern of our partner during the peak demand periods which motivated our study), gridlock probability, packer utilization and sorter utilization. However, it leaves aside the dimensions of order cycle time as well as storage and/or conveyor space 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 fully explore, because our models do not explicitly capture the detailed layout and resulting picking tours in the picking area (see §3.1 and related discussion in §6).

In order to enable a meaningful comparison with waveless policies, we consider models capturing the most sophisticated wave release policies observed in practice (see §1 for background), as described in §4.1 and §4.2 below.

4.1 Overlapping Waves Policy Under this policy and according to common industrial practice (see §1), complete waves of items corresponding to a number of orders equal to the number of sorter 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 §3.1 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[14], for a reason that will soon become clear. Note that in the waveless picking model of §3 induction capacity is only considered implicitly through the dependence of transit times on the congestion upstream of the sorter. While justified for the stationary steady-state associated with waveless release, that approach is not appropriate to model transient 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 short 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 retrieved from the actual dataset of flow information obtained from our partner. 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 §3.1.

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, Quiroz and Sharp (1988)) and Johnson and Lofgren (1994) however, we consider the more general release policy whereby each 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 to the possibility of gridlock, which is still characterized by the event $Y(\tau) + Z(\tau) > n$, with

[14] The waves do not include single item orders, which are packed through a separate process in the warehouses of our industrial partner.

$Y(\tau)$ 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 $\lim_{\tau\to\infty} \mathbb{P}(Y(\tau) + Z(\tau) > n) = (1-\alpha)\beta$ (see §3.3).

4.2 Split Sorter Policy As described in §1 and Ruben and Jacobs (1992), Russell and Meller (2003) and Perkins (2008), this alternative policy involves the sequential release of waves with a number of orders equal to half the number of chutes in the sorter, or $n/2$. The sorter chutes are split in two halves (each dedicated to a separate wave), packers give priority to completed orders in the oldest wave, and a new wave is released as soon as one of the two halves becomes empty. The model we use to represent this policy is the same as described in §4.1, except that the size of each wave is halved 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 $W/2$.

5 Numerical Experiments

The goal of our simulation study is to estimate and understand the relative performance of various order release control policies in the setting of our industrial partner's warehouses. We first review the policies considered and other experimental design issues in §5.1, then present and discuss our results in §5.2. The Online Appendix also contains additional experiments designed to assess the robustness of the waveless release policies considered with respect to transient disruptions.

5.1 Experimental Design In addition to the waveless release policy ADP^β derived in §3 and the wave-based release policies W_Ω and $W/2$ defined in §4.1 and §4.2 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). We consider CONWIP policies here because despite their simplicity, they have been found to perform well in many environments (Spearman and Zazanis 1992).

The range of simulation scenarii we consider is characterized by three different values for the number of packers $w \in \{p,\ 1.125p,\ 1.25p\}$, where p denotes the average number of packers working in our industrial partner's warehouse during a peak demand period, under the staffing policies prevailing at the beginning of our interaction. 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 §3.3). In practice the level of gridlock risk associated with $\bar{\beta}$ is deemed unacceptably high by our industrial partner, but we consider it here to perform an analysis of sensitivity relative to 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 (graphical representations of system behavior suggest that all policies have long reached steady-state by then)[15]. In order to enable a meaningful assessment and preserve our partner's confidential information, all throughput results are provided as a ratio to the average throughput γ^{HIST} observed in our industrial partner's warehouse during a period with no breaks and p packers assigned to the sorter. Note that for all policies the packing capacity $w(\mathrm{E}[C])^{-1}$ (or service rate of the third station described in §3.1 and §4) constitutes an upper bound for the throughput rate, 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

[15] Note that $\gamma^{CST} = \lambda^{CST}$.

	Number of Packers (w)	p	$1.125p$	$1.25p$
	Chutes per Packer (n/w)	55	48.8	44
$ADP^{\underline{\beta}}$ [$ADP^{\bar{\beta}}$]	Throughput	104.4 [104.4]	109.3 [109.5]	110.3 [110.4]
	Packing Utilization	99.4 [99.4]	92.6 [92.8]	84.1 [84.1]
	Sorter Utilization	85.5 [88.2]	78.2 [79]	80 [81.2]
$CWP^{\underline{\beta}}$ [$CWP^{\bar{\beta}}$]	Throughput	104 [104.3]	105.3 [106.4]	105.8 [106.5]
	Packing Utilization	99.0 [99.3]	89.2 [90.1]	80.6 [81.2]
	Sorter Utilization	80.4 [83.2]	77.1 [78.2]	77.1 [78.9]
$CST^{\underline{\beta}}$ [$CST^{\bar{\beta}}$]	Throughput	101.9 [103]	102.7 [104.6]	102.8 [104.7]
	Packing Utilization	97.0 [98.0]	87.0 [88.7]	78.3 [79.8]
	Sorter Utilization	74.3 [74.5]	74.1 [74.3]	74.9 [75.2]
$W_{\Omega^{\underline{\beta}}}$ [$W_{\Omega^{\bar{\beta}}}$]	Throughput	98.7 [101.5]	108.9 [109.9]	110.3 [110.6]
	Packing Utilization	94.7 [97.2]	92.6 [93.5]	84.0 [84.6]
	Sorter Utilization	78.1 [80.7]	79.9 [80.9]	76.7 [76.8]]
	$\Omega^{\underline{\beta}}$ [$\Omega^{\bar{\beta}}$]	49 [46]	44 [37]	31 [30]
W_{60}	Throughput	89.9	95.3	100
	Packing Utilization	86	81	76.5
	Sorter Utilization	70.7	69.9	69.3
	$\mathbb{P}$(Gridlock)	0	0	0
W_{100}	Throughput	66.2	71.3	75.9
	Packing Utilization	63.3	60.6	52.5
	Sorter Utilization	52.1	52.3	52.6
	$\mathbb{P}$(Gridlock)	0	0	0
$W/2$	Throughput	103.2	105.3	105.4
	Packing Utilization	98.7	89.6	80.7
	Sorter Utilization	61.6	38.2	36.2
	$\mathbb{P}$(Gridlock)	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.

an estimate obtained from simulation for the average number of busy chutes in steady-state under each policy D considered.

5.2 Results and Discussion Table 1 summarizes our steady-state simulation results. We discuss here separately the results for the waveless policies (in §5.2.1) then the results for the wave-based policies (in §5.2.2). A relative performance comparison between the two types of policies based on these results is provided as part of our concluding remarks in §6. We also refer the reader to the Online Appendix for a discussion of additional simulation experiments conducted to assess the robustness of some of the policies considered with respect to transient shocks and inacurrate input data.

5.2.1 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 the number of packers w is equal to p (and, a fortiori, when $w < p$), 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 the maximum release rate is substantially higher than the packing capacity, both in practice and in all simulation scenarios considered here, only the last two are relevant. With a relatively low number of packers $w \leq p$, packing is effectively the system bottleneck as the throughput of all policies remains relatively close then to the overall packing capacity. Because packing capacity is an upper bound on the long-term average 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, both their effective utilization and the marginal gain in throughput from this additional packing capacity decrease under all policies considered, so that the gridlock constraint 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 a set of $G/M/w$ queueing 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 (as the gridlock probability constraint requires). Another relevant insight from queueing theory is that the performance measures of highly congested queues are much 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 scenario where $w = p$. 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, and therefore greater throughput, for the same level of risk.

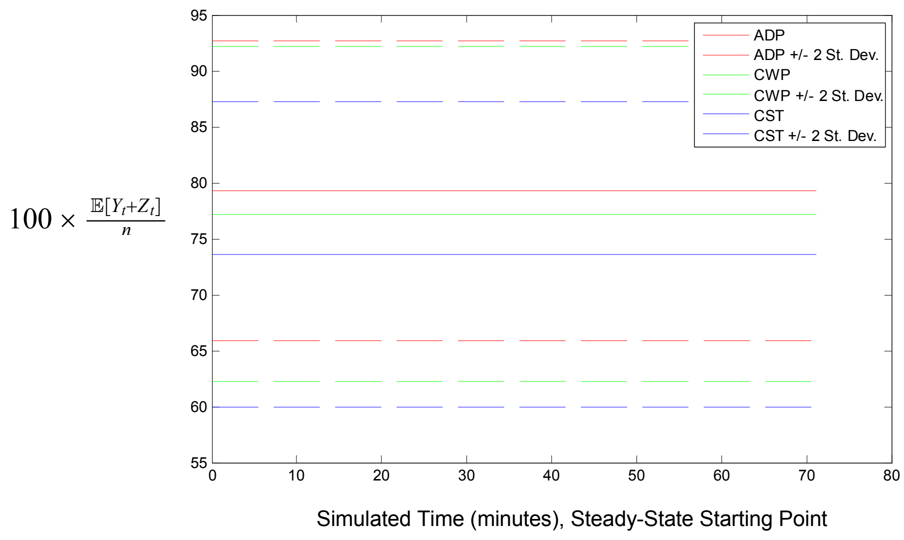


Figure 3: Two Standard Deviation Range of the Steady-State Number of Busy Chutes $\mathbb{E}[Y_t + Z_t] \pm 2\sigma[Y_t + Z_t]$ for Policies $ADP^{\underline{\beta}}$, $CWP^{\underline{\beta}}$ and $CST^{\underline{\beta}}$ with $w = 1.25p$ Packers.
Note: Statistics are computed after 2 days of simulated 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 is not coincidental, and in fact lends support to the validity of our results. The organizational structure used in our partner's warehouse defines a picking team and a packing team, with separate performance metrics. A key metric used for the packing team is the average worker productivity, defined for a given 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 the appropriate staffing level of packers should be determined by weighing the labor cost incurred against the overall system throughput it enables. In particular, 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, as surge packing capacity play an important role in avoiding gridlock. In that respect, the results shown in Table 1 should enable a more precise examination of this trade-off by our industrial partner, and 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 observed in our industrial partner's warehouse under comparable staffing conditions. It may be surprising at first that even policy CST^{β} outperforms the policy used in our partner's warehouse. We point out however that despite its simplicity CST^{β} is still obtained through optimization (over the constant release rate λ^{CST}), while the release policy used by our partner at the beginning of our interaction was not fully formalized and relied at least in part on the judgment of employees having sometimes little or no experience with warehouse dynamics during peak demand periods. Unfortunately, similar historical performance data was not readily available to us for numbers of staffed packers that are different than p. However, assuming that the relative performance of CST^{β} and our partner's historical policy would be maintained in such scenarii, 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 relative to the historical performance we have observed by about 10%.

5.2.2 Wave-Based Release Policies System dynamics under the wave-based policies W_Ω and $W/2$ are qualitatively very different from that under the waveless policies discussed in §5.2.1. As should be intuitively obvious from their definition in §4, these policies give rise to a periodic or cyclical steady-state, with a period equal to the average time between consecutive wave releases. To better 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 various wave-based policies of interest.

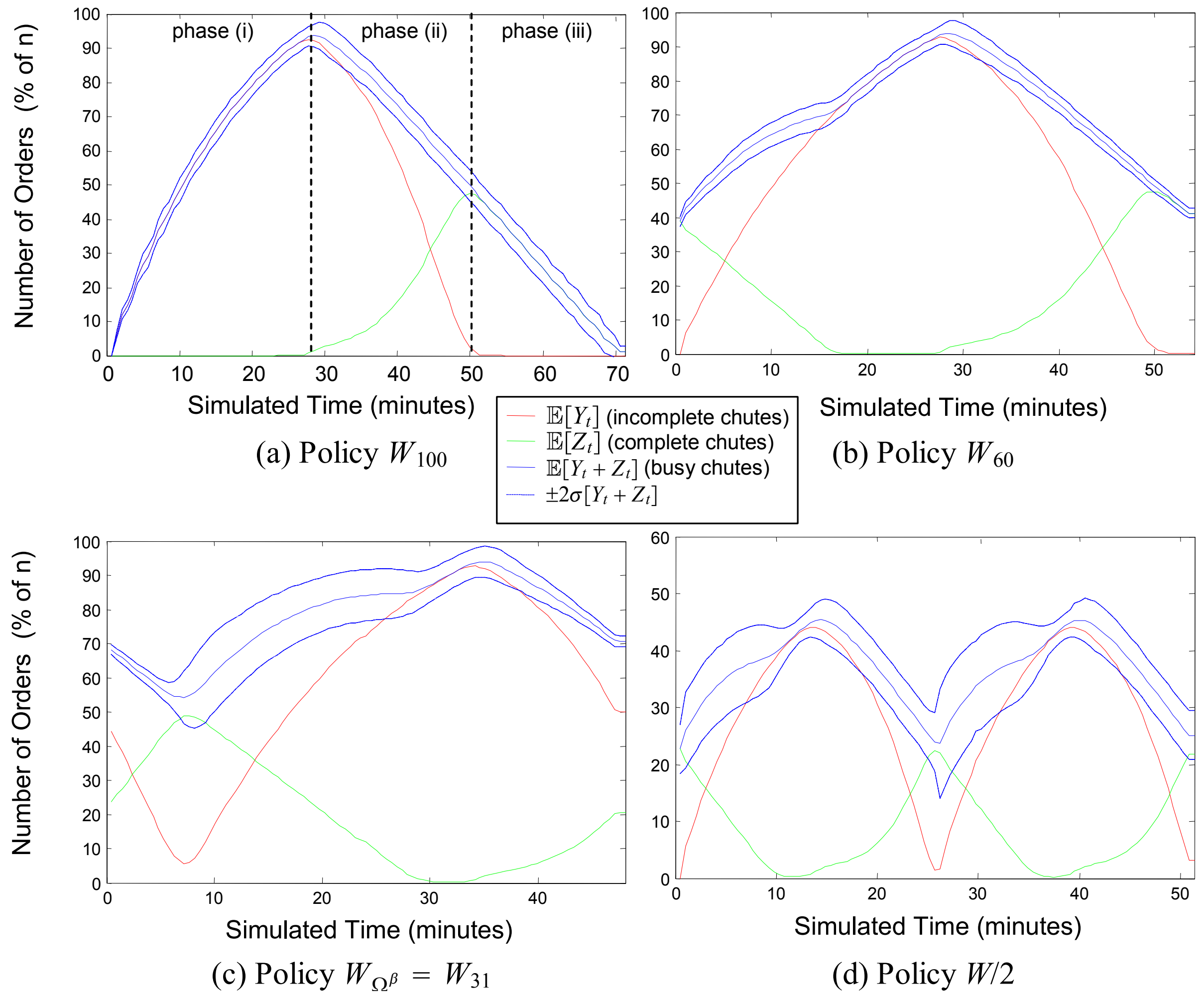


Figure 4: Periodic Cycles of the Average Number of Incomplete Chutes $\mathbb{E}[Y_t]$, Complete Chutes $\mathbb{E}[Z_t]$ and Busy Chutes $\mathbb{E}[Y_t+Z_t]\pm 2\sigma[Y_t+Z_t]$ in Steady-State for Selected Wave-Based Release Policies with $w = 1.25p$ Packers. Notes: Statistics are computed between 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.

The example of policy W_{100} illustrated by Figure 4 (a) is relevant because this simple form of wave release policy seems to be fairly common (see Armstrong, Cook and Saipe 1979, Meller 1997 and §1) and also because it is a useful starting point for understanding the other wave-based release policies. Its periodic cycle corresponds by definition 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 §1). 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[16]. As a result, the throughput of this policy is about 30% lower than that of some other policies tested, including some with no risk of gridlock, 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 observation suggests that overall throughput may be considerably increased with little additional risk of gridlock by processing in parallel two or more waves which have been appropriately staggered. Specifically, by releasing another wave late in the second phase (as W_{Ω^β} does) or in the third phase (as W_{60} does) shown in

[16] 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.

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 formal 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 ($W_{\Omega^{\beta}}$) associated with one wave effectively overlaps with the first phase of the next one. Figures 4 (a), (b) and (c) thus illustrate the evolution of system dynamics occurring when waves are increasingly made to overlap in this manner, and the shorter periods seen in their x-axis show the corresponding improvement in throughput (the number of orders released at the beginning of each period is equal to n for all policies W_{Ω} considered) – note also from these Figures that the average idling period of packers within each cycle increases with Ω. More specifically, it is seen in Figure 4 (b) (resp. (c)) that, consistent with the definition of W_{60} (resp. $W_{\underline{\Omega}^{\beta}}$), new waves are released under that policy when 40% (resp. $1 - \underline{\Omega}^{\beta} = 69\%$) of the chutes in the sorter are occupied. When the relatively mild degree of 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 corresponding to $\underline{\Omega}^{\beta} = 31$, 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. As a result, it takes some time on average for the induction stations to start processing items from the new wave after it is released, which explains why the number of busy chutes initially decreases at the beginning of the cycle – note from Figure 4 (c) that any given wave only completes packing approximately 30 minutes after the subsequent one is released.

In addition, because the estimated gridlock probability for W_{60} reported in Table 1 is zero across all scenarios, that policy illustrates the existence of a range for the overlap parameter Ω where there is no trade-off between throughput performance and risk. However, policy $W_{\underline{\Omega}^{\beta}}$ 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 $\underline{\Omega}^{\beta} = 49$, 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 two or more 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 $W/2$ which are illustrated by Figure 4 (d) are easily understood by noting that each half of the sorter processes then non-overlapping waves of size $n/2$. However, waves assigned to different sorter halves do overlap, creating dynamics for the overall number of incomplete, complete and busy chutes which are qualitatively comparable to that observed under W_{60}. However, the peaks of the number of busy chutes under $W/2$ occur about twice as frequently, and only involve one half of the sorter each time. As seen in Table 1, these smaller and more frequent releases result in relatively high packing utilization (80.7% with $1.25p$ packers), and therefore a throughput performance only slightly lower to that of $W_{\Omega^{\underline{\beta}}}$, which is particularly remarkable because $W/2$, in contrast with $W_{\Omega^{\underline{\beta}}}$, 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 §1 and following discussion in §6).

6 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 operational control guidelines for waveless picking during peak demand periods, with the goal of maximizing throughput while keeping the likelihood of gridlock sufficiently low.* We presented in §3 a queueing model of our partner's waveless operation which is adapted to the input data available in practice and describes the most important process flow dynamics in that setting. This model seems to have appropriate predictive accuracy (as established through a validation experiment described in §3.2.2) and may be embedded in an optimization formulation for which an approximate solution procedure can be used, as described in §3.3 and §3.4. This amounts to an operational solution to the problem defined in the objective statement. From the standpoint of improving our partner's waveless operation, our simulation results (in §5) suggest that its throughput may be increased by several percentage points with that solution (ADP), which is significant in this particular context given the high demand seasonality, the high cost of surge capacity, and the time-sensitivity of customers during peak season (e.g. Christmas). They also indicate that packing was a bottleneck for our partner, presumably because of local staffing incentives which ignored the system-wide importance of surge packing capacity for avoiding gridlock. In addition, the benefits of using ADP relative to much simpler waveless release heuristics such as CONWIP (CWP) and constant release (CST) increase substantially with more packing capacity (see §5.2.1). In any case, a salient quantitative prediction from this study is that an increase of staffed packers by 25% along with the implementation of the policy ADP we derived could increase process throughput by as much as 10%.

(Objective 2) *Leverage this model to conduct a rigorous performance comparison between wave-based and waveless release policies in the context of our industrial partner's warehouses.* Slight modifications of the waveless model described above allowed us to simulate and understand the performance of the most common wave-based release policies found in practice (see §4 and §5.2.2). Because this wave release model relies on the same detailed input dataset used to simulate waveless release policies, it enables a meaningful comparison between these two different types of control. As is obvious from contrasting Figures 3 and 4, our results illustrate that the qualitative behavior of the workload seen by the sorter and the packers over time is strikingly different under wave-based and waveless policies. Specifically, wave-based policies give rise to contrasted periodic cycles featuring high predictable variability, while 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, which is obtained through a simulation-based search over the wave overlap parameter Ω so that both policies have the same gridlock probability. The throughput superiority of ADP^{β} over $W_{\Omega^{\beta}}$ increases when packing capacity is more constrained however, because the extent to which waves can overlap while still satisfying the gridlock probability constraint is reduced then, so that the relative ability of $W_{\Omega^{\beta}}$ to utilize packers is diminished (see Figure 4 (a)-(c) and associated discussion in §5.2.2). In addition, the comparison with the other wave-based policies considered (W_{100},

W_{60}, $W/2$) along the dimension of throughput is very favorable to ADP^{β} regardless of packing capacity. In particular, the performance differential relative to the non-overlapping wave policy W_{100} (more than 35% in all scenarios considered) is remarkable in light of that policy's prevalence – the deterministic assurance of never experiencing gridlock has therefore an enormous cost in terms of throughput when implemented through W_{100}. It should be noted however that these evaluations of throughput performance rely on a key assumption of our wave-based release model, which is that a complete wave of picked items is always available to be released into the sorter when required. That assumption is very salient with respect to our performance comparison, because it implies that all wave-based policies considered require considerably more buffer inventory and storage space upstream of the sorter (which requires either storage conveyors or handling labor) than waveless policies, and therefore experience longer order cycle times. In addition, should that assumption not be always quite satisfied in practice, the throughput performance of the wave-based release policies could be much lower than predicted by our model.

More generally our quantitative results, which only focus on selected performance dimensions, inform a broader qualitative comparison between wave-based and waveless release policies. To the extent that the model and procedure developed in §3 generate waveless release policies with an arbitrarily low probability of gridlock, our work reduces the relevance of that performance dimension to the comparison at stake. Although we find that policies ADP^{β} and $W_{\Omega^{\beta}}$ yield the same throughput when many packers are available, in practice very few facilities using wave-based release control seem to use an optimal value of the wave overlap parameter. As a result, many of these warehouses could possibly increase their overall processing capacity through waveless picking. Beyond throughput, waveless policies seem unquestionably superior along the dimensions of order cycle time, storage space and work-in-process/buffer inventory (see above discussion). On the other hand, a significant advantage of wave-based release policies which should not be underestimated in practice is their simplicity and low implementation cost. The split-sorter policy $W/2$ is remarkable in that respect, because it does not require the determination of any parameter or other preliminary computation, yet performs relatively well in terms of throughput across all scenarios considered (which is unsuprising given it has been designed to achieve a high packing utilization, see §1). In terms of software implementation, many vendors offer so-called warehouse management systems (WMS) or warehouse control systems (WCS), which among other functionalities facilitate the management of wave-based release policies. In contrast, software solutions supporting waveless picking are not yet as widely available, and seem to be occasionally developed internally. Finally, a 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 §3, such as storage policies and the layout of the order picking area. We point out however that waveless picking does seem a priori to have very strong advantages in that regard. Specifically and as discussed in §1, 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 for separating items belonging to different waves before induction.

In closing, we observe that given order cycle times are becoming increasingly important competitive factors and the costs of technologies and tools required by waveless policies seem likely to continue decreasing in the future, waveless picking appears worthy of serious consideration for any warehouse with an automated sorter, particularly for any new construction project which does not involve switching costs and resistance to change. Our work may also have short-term implications for existing warehouses using a non-overlapping wave release policy similar to W_{100}. Specifically, facilities among these where the layout of the order picking area, the number of skus and the number of sorter lanes n are such that using waves of size $n/2$ would not significantly reduce picking labor productivity should consider switching to a split sorter policy $W/2$. Also, facilities with less margin in terms of wave size should consider overlapping waves, as our results suggest the existence of a fairly large range of values for the wave overlap parameter which entail no risk of gridlock. These recommendations are clearly tentative however, because they are only supported by a study of a few specific warehouses belonging to a single firm.

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 Steve Graves, Gabriel Bitran, Jeffrey

Wilke, Russell Allgor, Louis Usarzewicz, Brent Beabout, Dean Kassmann, Irina Aronova, Robert Abbondanzza, Mouin Sayegh, Felix Anthony, Nadia Shouraboura, Eric Young, Francois 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*.

Bragg, S. J. (2003). Analysis of sorting techniques in customer fulfillment centers. Master's thesis, Massachusetts Institute of Technology.

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.

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.

Forger, G. (2005, January). Joltin' java DC. *Modern Materials Handling*.

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*.

Gu, J., M. Goetschalckx, and L. F. McGinnis (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research 177*, 1–21.

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*.

Hordijk, A. and F. Spieksma (1989). Constrained admission control to a queueing system. *Advances in Applied Probability 21*, 409–431.

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 performance. *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.*

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. (2007). Continuous processing using a sorter. 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.

Spearman, M. L. and M. A. Zazanis (1992). Push and pull production systems: Issues and comparisons. *Operations Research 40*(3), 521–532.

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.

Online Appendix to: To Wave Or Not To Wave? Order Release Policies for Warehouses with an Automated Sorter

Jérémie Gallien[1] and Théophane Weber[2]

May 14, 2007

A.1. Warehouse Outbound Process Description

The warehouses which we focused on with our partner have an automated sorter with standard tray-tilting technology and a physical layout similar in key aspects to those described in the papers discussed in section §2 of the paper. We focus here on the outbound process which includes picking, sorting and packing, and leave aside the inbound operation of receiving and stowing. Figure A.1 provides a schematic layout representation.

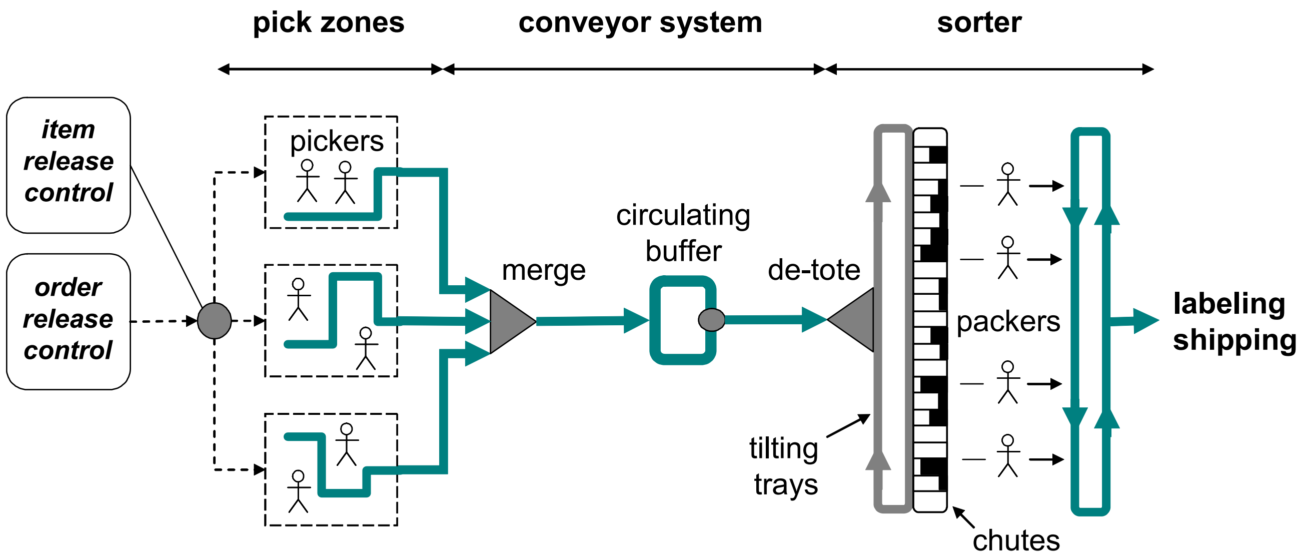


Figure A.1: Flow Diagram of the Pick-to-Ship Process Considered

Since our partner is an online retailer shipping directly to customers from its warehouses,

[1] Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02142. E-mail: jgallien@mit.edu
[2] Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02142. E-mail: theo_w@mit.edu

a first noteworthy feature is the use of split-case picking and sorting, as in Johnson and Meller (2002). Also, the very large selection of items offered by our partner (a marketing advantage often leveraged by online retailers, who are not limited by the page and shelf space restrictions of mail order catalogs and physical stores, respectively) results in a relatively large picking area and long item travel times to the sorter. As in many other large warehouses, the picking area is subdivided into several zones, each having its dedicated team of workers (*pickers*). Pickers have portable digital 2-way wireless communication devices with a bar code scanner and an LCD screen showing the nature and location of items to be picked. Picking an item involves scanning its bar-code and placing it into one of several plastic containers (*totes*) carried by an individual rolling cart. Totes are offloaded when full onto a conveyor belt spreading through their pick zone, which relieves the pickers from unloading travel, as described in Owyong and Yih (2006). Conveyor belts carrying totes coming out of all the pick zones lead after a merge point to an accumulation buffer where selected totes may be temporarily held for the purpose of reducing the accumulation time of orders in sorter chutes (*chute-dwell time*), or time between the arrivals of the first and last item of each order in a chute. In our partner's warehouses, this accumulation conveyor upstream of the induction stations and the recirculating conveyor of tilting trays in the sorter is itself a recirculating loop (hereafter denoted *recirculating buffer*), as in Le-Duc and de Koster (2005). The induction (*de-tote*) stations have automated coordinated induction belts and were designed using realistic throughput models of the type described in Johnson and Meller (2002), resulting in relatively high capacity and low labor costs. In this setting, packers thus constitute the other large labor category of the outbound process besides pickers. They are tasked with putting the items from any completed chute into a cardboard box of appropriate size and place it onto a conveyor leading to automated stuffing and labeling stations. Their work is guided by a light system signaling every chute as complete (green), incomplete (orange), or unassigned (no light). Finally, we point out that our partner's warehouses use a sophisticated data collection system involving bar-code scanners carried by pickers and packers and also placed in many locations in the conveyor system, induction stations and sorter chutes. This system generates a database of detailed flow timing information for individual orders which provided many insights about the actual behavior of this process, as discussed in the next section.

A.2. Flow Data Analysis.

The database of order flow event timing mentioned in the previous section enabled a quantitative analysis of order flows in our partner's warehouses, and ultimately provided us with the distributional input data required by the quantitative models defined in sections §3.1 and §4 of the paper. A first quantity of interest that we analyzed is the empirical distribution of *transit time*, or time necessary for a given item to travel from the pick zone where it is collected to its assigned chute in the sorter. As an illustration, Figure A.2 shows the empirical p.d.f. of transit times for items picked from a given picking zone over a 24 hour period during the peak of the 2003 season, which constitutes a representative example of the many other such distributions we have constructed.

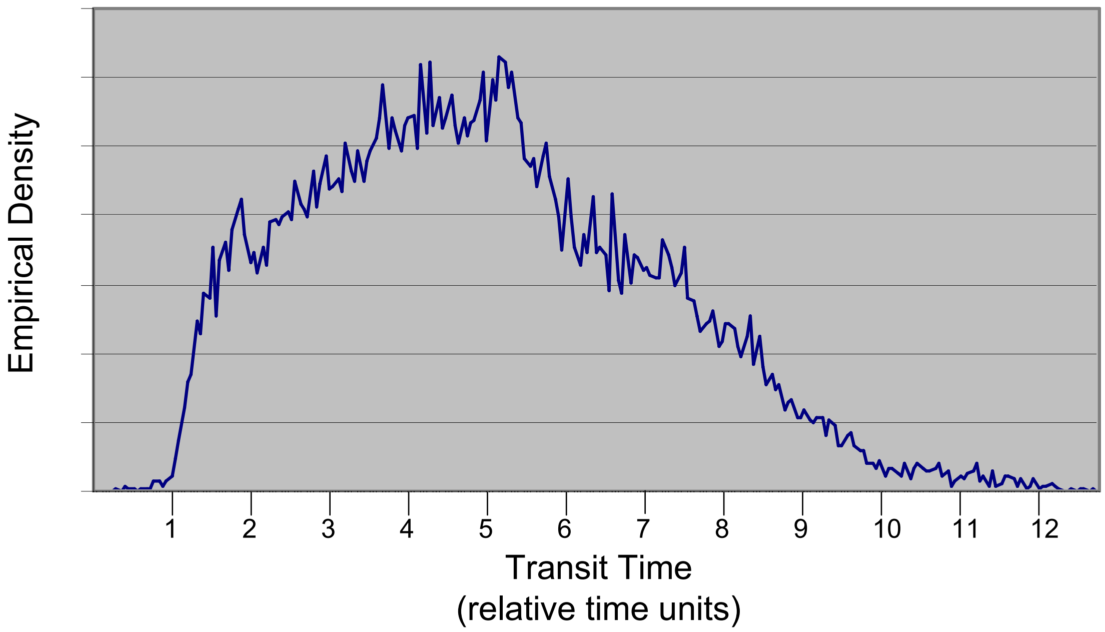


Figure A.2: Empirical Density of Transit Times from the Same Picking Zone Over 24 Hours

Note that the relative time units used for the x-axis in Figure A.2 to disguise our partner's confidential data show a variation from 1 to 12 of the item transit times over that period, which was typical across all picking zones over that time period. Another typical feature is that the distribution shown in Figure A.2 is multi-modal, suggesting that it results from the superposition of several heterogeneous system behavior modes. We hypothesized that these behavioral modes were primarily driven by conveyor congestion, and that the overall behavior or the pick-to-ship process could be characterized fairly accurately using a limited number of congestion levels, each corresponding to a range of values for the total number of items on the conveyor system between the picking area and the sorter. To verify that hypothesis, we constructed and plotted the data series representing the number of items on the entire conveyor system over time during the same 24h period, and we defined a limited number of congestion levels based on the amount of data available (indexed in the following as $g \in \{1, ..., \bar{g}\}$). Figure A.3 illustrates this process on a dataset which led us to define 7 congestion levels.

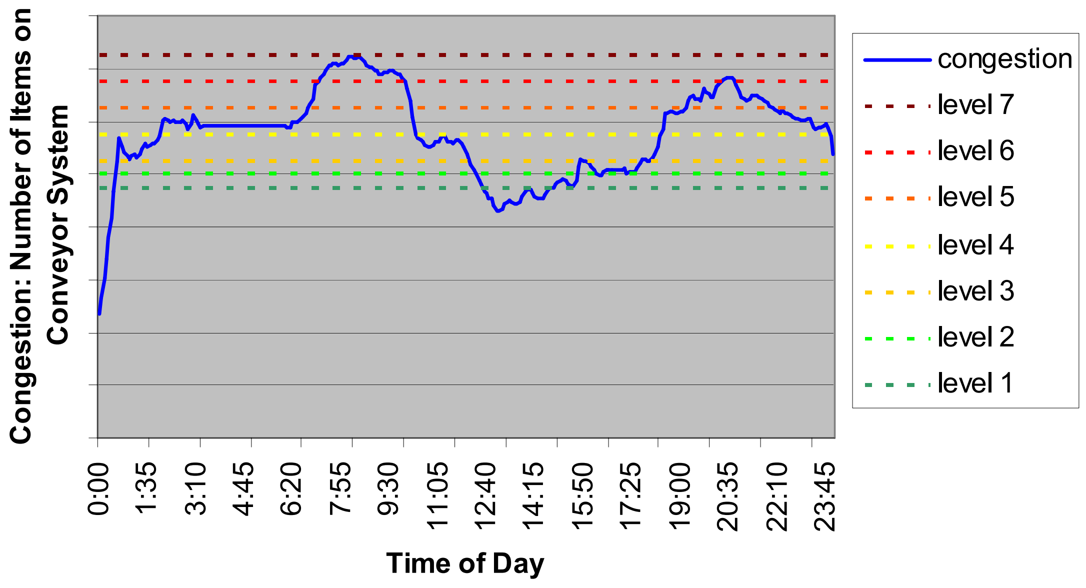


Figure A.3: Number of Items on Conveyor System Over 24 Hours

The next step was to construct the empirical item transit time distributions for each picking zone again, but this time for each congestion level separately. That is, instead of considering all the items picked from a given picking zone over 24 hours as before, we only considered the items that were picked from that picking zone during the times when the system was in a given congestion level. Figure A.4 shows two such transit time distributions for the same conveyor zone as Figure A.2, and corresponding to congestion levels $g = 2$ and 6 respectively. They also show the Gumbel (or $CMT1$) distributions with the same first two moments as the empirical distributions just defined.

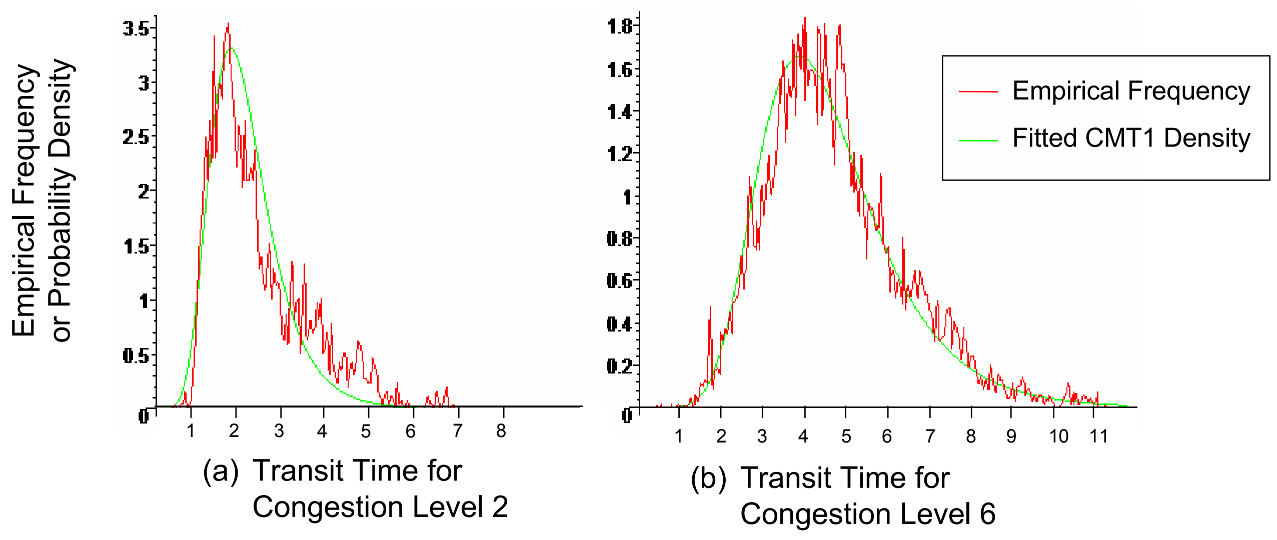


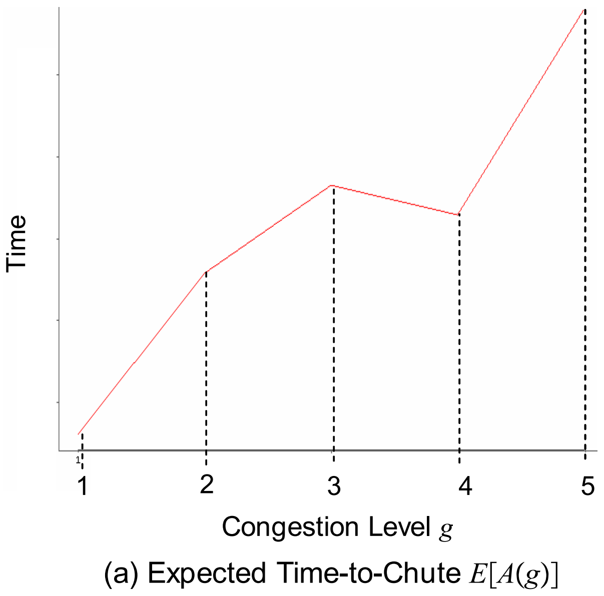
Figure A.4: Empirical Frequency and Fitted $CMT1$ Density for Transit Times from a Conveyor Zone Over 24 Hours for Congestion Levels $g = 2$ and $g = 6$

These figures illustrate the following features, which we found typical of all other picking zones and congestion levels:

- These empirical transit time distributions seem (mostly) unimodal, seemingly validating at a qualitative level the hypothesis formulated earlier that the heterogeneous behaviors of transit times when observed over long periods of time can be satisfactorily explained by the variations of the system congestion level. Observe that the modes of the distributions represented in Figures A.4 (a) and (b) correspond exactly to the peaks observed on the distribution represented in Figure A.2 around the relative time values 2 and 4.5 respectively.

- The empirical transit time distributions seem to be very well fitted by $CMT1$ distributions. This is remarkable as the $CMT1$ distributions are typically only introduced because of their mathematical properties (as is the case in section §A.6 of this Online Appendix), and not their modeling potential. However, it has already been observed (Gallien and Wein 2001) that $CMT1$ distributions are suitable for modeling transportation times, as their sharp left tail represents typical physical limitations of the transportation means (in the present setting, the conveyor belt speed), while their heavier right tail accounts for all the potential problems encountered along the way (here, congestion at the merge points and delays at the circulation buffer for example).

Besides its predictive validity, the notion of congestion levels just defined also generated interesting new insights about the behavior of the pick-to-ship process, as shown by examining the dependence of the empirical time-to-chute and chute-dwell time distributions defined in section §3.1 of the paper on the congestion levels $g \in \{1, ..., \bar{g}\}$. As illustrated by Figure A.5 (a representative example constructed with 5 congestion levels), the mean time-to-chute $\mathbb{E}[A(g)]$ follows an unsurprising overall increasing trend with g, however the mean chute-dwell time $\mathbb{E}[B(g)]$ exhibits a noticeable drop at an intermediary congestion level, and increases beyond that. Our industrial partner and we believe this phenomenon, which we consistently



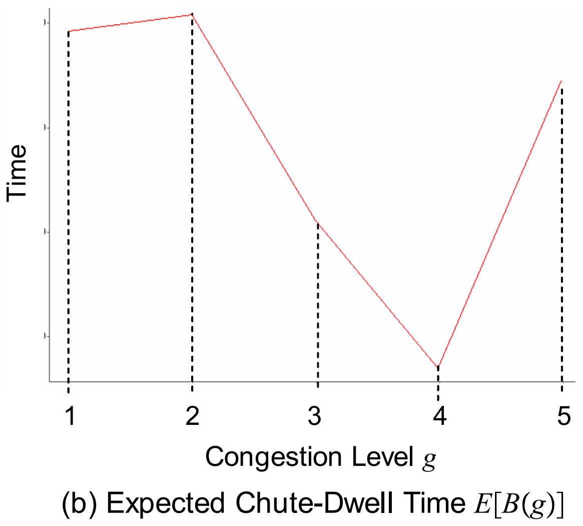


Figure A.5: Variation of Expected Time-to-Chute $E[A(g)]$ and Chute-Dwell Time $E[B(g)]$ with Congestion Level g

observed on several disjoint data sets and with various congestion level definitions, to be explained by the circulating buffer (see §A.1). Specifically, this buffer includes an active tote release logic allowing to dynamically delay the arrival of selected totes to the sorter, with the goal of reducing chute-dwell time for the orders containing items in those totes. The tote delaying logic implemented (which we are not at liberty to describe in more details) does not have any impact for low congestion levels (such as 1 and 2 on Figure A.5 (b)). For medium to high congestion levels (3 and 4 on Figure A.5 (b)) however, the circulating buffer performs its function adequately and the active control logic implemented results in a significant reduction of the average order chute-dwell time. This buffer does have a limited capacity however, so that when congestion increases further (to 5 on Figure A.5 (b)), it becomes full and, in order to preserve throughput, loses then its ability to increase the sojourn time of selected totes (think of Little's law). The role played by this buffer also explains the slight drop of $\mathbb{E}[A(g)]$ observed in Figure A.5 (a) at congestion level 4, although this is not nearly as significant. In summary, this data analysis uncovered the existence of a non-trivial nominal operating regime (congestion level 4 in Figure A.5) resulting from the design of this process, further motivating the development of order release control policies able to stabilize the process around it (see section §3.5 in the paper).

A.3. Statement and Discussion of Search Algorithm.

We now describe the search algorithm that we implemented in order to compute approximations of the multiplier θ and policy $\boldsymbol{\lambda}^{\theta}$ solving both $UDP[\theta]$ and $CDP[\beta]$, as described in section §3.4 of the paper:

Algorithm SEARCH[ϵ] **input:** Input data for problem $CDP[\beta]$, numbers $\underline{\theta}, \bar{\theta} \geq 0$ such that $\underline{\theta} \leq \theta^* \leq \bar{\theta}$ with θ^* defined as in Lemma 1 in the paper.

output: A number θ and policy $\boldsymbol{\lambda}^{\theta}$ that is near optimal for $CDP[\beta]$.

```
1. Set $k = 1$, $\underline{\theta}^k = \underline{\theta}$, $\overline{\theta}^k = \bar{\theta}$;
2. Set $\theta^k = \frac{\underline{\theta}^k + \overline{\theta}^k}{2}$; compute an optimal solution $\boldsymbol{\lambda}^{\theta^k}$ to $UDP[\theta^k]$, and $\mathbf{c}^{\theta^k}(x, y, z)$;
3. If $\mathbf{c}^{\theta^k}(x, y, z) > \beta$ set $\underline{\theta}^{k+1} = \theta^k$ and $\overline{\theta}^{k+1} = \overline{\theta}^k$; otherwise set $\underline{\theta}^{k+1} = \underline{\theta}^k$ and
   $\overline{\theta}^{k+1} = \theta^k$;
4. If $(\overline{\theta}^{k+1} - \underline{\theta}^{k+1}) < \epsilon$, stop, set $\theta^f = \overline{\theta}^{k+1}$, compute an optimal solution $\lambda^f$ to $UDP[\theta^f]$,
   and return $(\theta^f, \boldsymbol{\lambda}^f)$; otherwise set $k = k + 1$ and go to step 2.
```

We initialized that algorithm by setting $\underline{\theta}$ to 0 and $\bar{\theta}$ to the first value of a geometric sequence $(\theta^k)_{k \in \mathbb{N}}$ such that $\mathbf{c}^{\theta^k}(x, y, z) > \beta$. Note that, as described in the paper, we only compute approximate solutions to the unconstrained DPs $UDP[\theta^k]$ stated in the algorithm definition. While we were not able to develop a theoretical characterization of the convergence properties of SEARCH[ϵ], the following Lemma provides an a posteriori bound for the suboptimality of any policy to which it converges:

Lemma 1 *Let (x, y, z) be the initial system state, let (λ, θ) be the output of algorithm* SEARCH[ϵ]*, and let λ^* an optimal policy for $CDP[\beta]$. Then $c_{\lambda}(x, y, z) \leq \beta$, i.e. λ is* *feasible for $CDP[\beta]$, and*

$$r_{\lambda^*}(x,y,z) - \theta(\beta - c_\lambda(x,y,z)) \leq r_\lambda(x,y,z) \leq r_{\lambda^*}(x,y,z). \tag{A.1}$$

Proof: Let (θ^*, λ^*) be as defined in the statement of Lemma 2 in the paper, and let (θ, λ) be the algorithm output. Because $\theta > \theta^*$, from Lemma 3 in the paper we have $r_\lambda(x,y,z) \leq r_{\lambda^*}(x,y,z)$ and $c_\lambda(x,y,z) \leq c_{\lambda^*}(x,y,z) = \beta$, hence λ is feasible. Furthermore, since the policy λ^* is suboptimal for $UDP[\theta]$,

$$r_{\lambda^*}(x,y,z) - \theta.c_{\lambda^*}(x,y,z) \leq r_\lambda(x,y,z) - \theta.c_\lambda(x,y,z).$$

Substitution yields (A.1), completing the proof.

The intuitive interpretation for the suboptimality gap $\theta(\beta - c_\lambda(x,y,z))$ appearing in (A.1) is that suboptimality increases when the final policy λ does not use all the allowed risk provided by the model formulation, and this effect is all the more sensitive as the penalty for violating the constraint is high. That gap however can indeed only be evaluated a posteriori, since neither the final value of θ nor the difference $\beta - c_\lambda(x,y,z)$ are known in advance. The missing link in this characterization of convergence properties is a relationship showing that (and how) $\theta(\beta - c_\lambda(x,y,z))$ decreases as ϵ goes to zero, which we have unfortunately not been able to establish theoretically. In practice however, the final value of $\theta(\beta - c_\lambda(x,y,z))$ obtained for our choice of ϵ was always less than 2% (and in most cases, less than 1%) of $r_{\lambda^*}(x,y,z)$. This is not surprising because from Lemma 1 in the paper, there exists an optimal policy for $CDP[\beta]$ which only randomizes between two actions in one state; given the very high number of states, randomization in a single one has little impact, and deterministic policies can match the desired risk value for all practical purposes.

A.4. Description of Approximate DP Algorithm.

We now describe the algorithm that we have implemented in order to solve approximately each instance of the dynamic program $UDP[\theta]$ described in section §3.4 of the paper and the previous section of this Appendix[3]; additional background on the corresponding approximate dynamic programming methods and concepts may be found in Bertsekas and Tsit-

[3] For notational simplicity, we omit any dependence on θ of the functions and variables mentioned in this subsection.

siklis (1996). Our first approximation consists of discretizing the state and control spaces. Specifically, we consider increasing finite sequences $\hat{\mathbf{x}} = \{\hat{x}_i\}_{i=0}^m$, $\hat{\mathbf{y}} = \{\hat{y}_j\}_{j=0}^m$, $\hat{\mathbf{z}} = \{\hat{z}_k\}_{k=0}^m$, $\hat{\boldsymbol{\lambda}} = \{\hat{\lambda}_i\}_{i=0}^{\ell}$ and the projection $\mathcal{P} : \mathbb{N}^3 \to \hat{\mathbf{x}}\times\, \hat{\mathbf{y}}\times\, \hat{\mathbf{z}}$ defined such that $\mathcal{P}(x, y, z)$ minimizes within $\hat{\mathbf{x}}\times\, \hat{\mathbf{y}}\times\, \hat{\mathbf{z}}$ the rectangular distance to state $(x, y, z) \in \mathbb{N}^3$. The control space $\hat{\boldsymbol{\lambda}}$ is obtained by a regular discretization $\hat{\lambda}_i \triangleq i\bar{\lambda}/\ell$, but our state space discretization is denser around the states that are more likely to be visited often. That is, $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ are constructed such that the simulated steady state occupancy measure $\mathbb{P}(\mathcal{P}^{-1}(\hat{x}_i, \hat{y}_j, \hat{z}_k))$ under the best constant solution to the optimization problem $CDP[\beta]$ defined in the paper is approximately constant over (i, j, k), subject to a maximum value constraint for the discretization step sizes $\hat{x}_{i+1} - \hat{x}_i$, $\hat{y}_{j+1} - \hat{y}_j$ and $\hat{z}_{k+1} - \hat{z}_k$.

Secondly, we implement a policy iteration algorithm relying on an approximate Robbins-Monro stochastic approximation scheme for the evaluation step, and Monte-Carlo simulations for the improvement step. Starting with a policy $\boldsymbol{\lambda}^q : \hat{\mathbf{x}}\times\, \hat{\mathbf{y}}\times\, \hat{\mathbf{z}} \to \hat{\boldsymbol{\lambda}}$ and initial value function estimates j_q^0, r_q^0 and c_q^0 defined over $\hat{\mathbf{x}}\times\, \hat{\mathbf{y}}\times\, \hat{\mathbf{z}}$, the evaluation step implements the recursion

$$\left\{\begin{array}{l} r_q^{s+1}(\hat{x}, \hat{y}, \hat{z}) = (1-\gamma_s) r_q^s(\hat{x}, \hat{y}, \hat{z}) + \gamma_s \left[\lambda^q(\hat{x}, \hat{y}, \hat{z}) + \alpha . r_q^s(\mathcal{P}(x', y', z'))\right] \\ c_q^{s+1}(\hat{x}, \hat{y}, \hat{z}) = (1-\gamma_s) c_q^s(\hat{x}, \hat{y}, \hat{z}) + \gamma_s \left[1_{\{\hat{y}+\hat{z}>n\}} + \alpha . c_q^s(\mathcal{P}(x', y', z'))\right] \\ j_q^{s+1}(\hat{x}, \hat{y}, \hat{z}) = r_q^{s+1}(\hat{x}, \hat{y}, \hat{z}) - \theta . c_q^{s+1}(\hat{x}, \hat{y}, \hat{z}), \end{array}\right. \tag{A.2}$$

for all $(\hat{x}, \hat{y}, \hat{z}) \in \hat{\mathbf{x}}\times\, \hat{\mathbf{y}}\times\, \hat{\mathbf{z}}$, where $\gamma_s \triangleq \frac{a}{b+s}$ is a diminishing step function (a and b are constant), and (x', y', z') denote a simulated realization under system (4) in the paper of variables $(X_{t+1}, Y_{t+1}, Z_{t+1})$ given $(X_t, Y_t, Z_t) = (\hat{x}, \hat{y}, \hat{z})$ and $\lambda_t = \lambda^q(\hat{x}, \hat{y}, \hat{z})$. Termination for recursion (A.2) is triggered by either $s + 1 = n_{\text{eval}}$ or

$$\sup_{(\hat{x},\hat{y},\hat{z})} |j^{s+1}(\hat{x}, \hat{y}, \hat{z}) - j^s(\hat{x}, \hat{y}, \hat{z})| \leq \epsilon_1,$$

where n_{eval} is a specified maximum number of policy evaluation steps and $\epsilon_1 > 0$ is a specificed accuracy parameter. At that point, $\mathbf{j}_q^{s+1}$ is considered an estimate for the value function $\mathbf{j}_q$ of policy $\boldsymbol{\lambda}^q$. The ensuing policy improvement step consists of computing

$$\lambda^{q+1}(\hat{x}, \hat{y}, \hat{z}) = \arg\max_{\lambda \in \hat{\boldsymbol{\lambda}}} \left(\lambda - \theta . 1_{\{\hat{y}+\hat{z}>n\}} + \alpha \frac{1}{n_{\text{mc}}} \sum_{\omega=1}^{n_{\text{mc}}} \mathbf{j}_q(\mathcal{P}(x'_\omega, y'_\omega, z'_\omega)) \right) \tag{A.3}$$

for all $(\hat{x}, \hat{y}, \hat{z}) \in \hat{\mathbf{x}}\times\, \hat{\mathbf{y}}\times\, \hat{\mathbf{z}}$, where $(x'_\omega, y'_\omega, z'_\omega)_{\omega=1}^u$ are n_{mc} simulated realizations under system (4) in the paper of variables $(X_{t+1}, Y_{t+1}, Z_{t+1})$ given $(X_t, Y_t, Z_t) = (\hat{x}, \hat{y}, \hat{z})$ and $\lambda_t = \lambda$. The evaluation step (A.2) applied to policy $\boldsymbol{\lambda}^{q+1}$ provides then an estimate for its value function

$\mathbf{j}_{q+1}$. At that point the main recursion loop just described is repeated (and the algorithm proceeds to another policy improvement step), unless $q+1 = n_{\text{improv}}$ or

$$\sup_{(\hat{x},\hat{y},\hat{z})} |j_{q+1}(\hat{x},\hat{y},\hat{z}) - j_q(\hat{x},\hat{y},\hat{z})| \leq \epsilon_2, \tag{A.4}$$

where n_{improv} is a specified maximum number of policy improvement steps and $\epsilon_2 > 0$ is a specificed accuracy parameter.

The computational time of the algorithm just described is primarily driven by the number of simulations of system (4) in the paper that it performs, which is bounded from above by

$$n_{\text{improv}}.m^3(\ell n_{\text{mc}} + n_{\text{eval}}).$$

In our numerical experiments, we have found that with about 22,000 states (m^3) and 100 control values (ℓ), the maximum number of improvement steps n_{improv}, evaluation steps n_{eval} and Monte-Carlo estimations n_{mc} could be chosen so that the final value of the l.h.s of (A.4), also known as the Bellman error, was no larger than 2% of the average value function upon algorithm termination. This required a computational time of about 30 minutes on a modern computer. Longer computations did lower the Bellman error further, but we observed that the resulting policy remained almost identical beyond that point. Finally, when implementing the dichotomic search over the multiplier θ described in section §3.4 of the paper and §A.3, we found that after solving 8 to 10 instances of $UDP[\theta]$ (or about 4 to 5 hours of computations) our suboptimality bound for the resulting policy relative to problem $CDP[\beta]$ was always below 2% (see Lemma 1).

A.5. Transient Robustness Experiments.

The goal of the set of simulation experiments reported here is to assess the robustness of the waveless picking policies considered in the paper relative to temporary mispecifications of the input data under which they are derived (see section §5 in the paper for a definition of notations and background on our simulation experiments). Such mispecifications may arise in practice as the result of undetected changes in process conditions, so that given the difficulty of monitoring such a large operation this issue is important to our industrial partner. Attempting to reproduce the actual process disruptions that we had most often heard about in various conversations with warehouse managers, we thus designed three transient simu-

	No Disruption	Experiment 1	Experiment 2	Experiment 3
$\mathbb{P}^{ADP}$(gridlock)	less than 10^{-4}	2.6	0.5	17
$\mathbb{P}^{CWP^{\beta}}$(gridlock)	less than 10^{-4}	49	39	6
$\mathbb{P}^{CST^{\beta}}$(gridlock)	less than 10^{-4}	5.7	47	4
$\mathbb{P}^{CWP^{\gamma}}$(gridlock)	7.6	100	68	42
$\mathbb{P}^{CST^{\gamma}}$(gridlock)	39	54	94	81

Table A.1: Gridlock Probabilities During Transient Simulation Experiments

lation experiments. All these simulated disruptions were initiated from steady-state (or 5 days of simulation under normal conditions) and with a number of packers equal to p, unless mentioned otherwise. We mostly considered the simulated response of the policies ADP, CWP and CST obtained for a risk level $\beta = \underline{\beta}$ assuming $w = p$ packers. For reasons that will soon be clear, the last two will be thereafter denoted by CWP^{β} and CST^{β}. Because both initial risk and throughput performances seem to provide an appropriate comparison basis, we also considered the policies CWP and CST obtained for $w = p$ and risk levels resulting in the same throughput for these policies as the throughput γ^{ADP} of the policy ADP obtained with $(w, \beta) = (p, \underline{\beta})$. These are noted here CWP^{γ} and CST^{γ}. The main performance metric that we monitored in these experiments is the proportion of simulation replications where the gridlock event $Y_t + Z_t > n$ did occur during the 2 simulated days following the start of the disruption, noted here $\mathbb{P}^{D}$(gridlock) where D is any of the policies $\{ADP, CWP^{\beta}, CST^{\beta}, CWP^{\gamma}, CST^{\gamma}\}$. Table A.1 contains a summary of our results[4], which we discuss in the remainder of this section after a more detailed description of each experiment.

A.5.1. Experiment 1: Conveyor Speed-Up Our first experiment consists of temporarily decreasing all time-to-chute (first station service times) by 20% during 6 hours. This design was motivated by the possibility in the actual system that the speed of one or several conveyor belts would increase above its normal value, or that the merge priority of a loaded conveyor belt relative to others would become temporarily high, triggering a faster release of items onto the sorter. Because some of these items would be chute openers (first items of an order), this can result in a sharp increase of the number of busy chutes, potentially leading

[4] Table A.1 notes: All results are shown as percentages, and have a standard estimation error from simulation lower than 0.5%.

to gridlock.

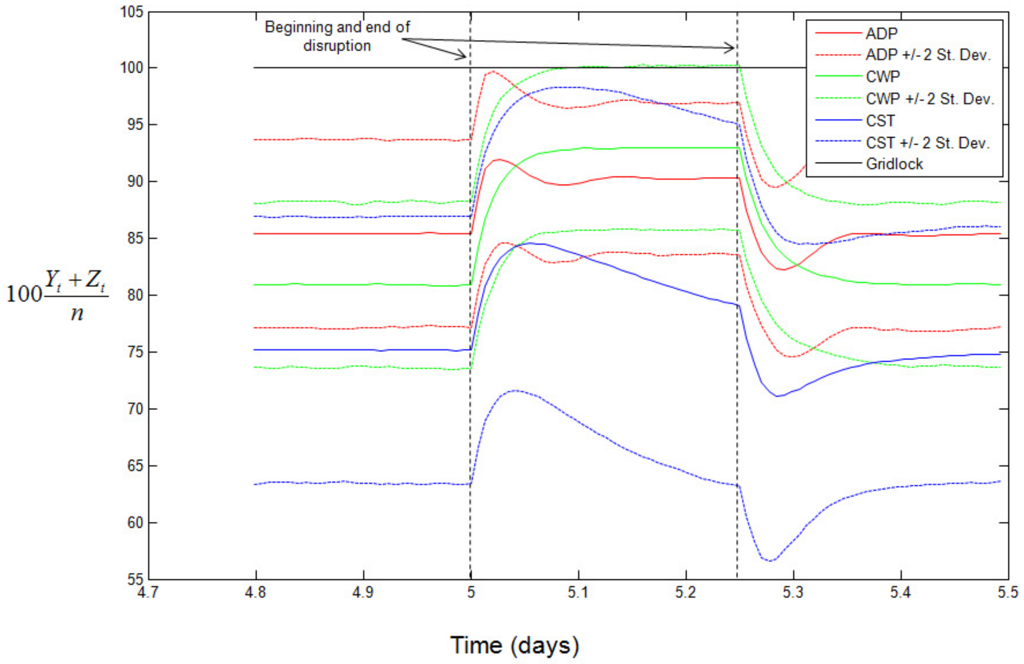


Figure A.6: Evolution of the Average Fraction of Busy Chutes $\frac{Y_t+Z_t}{n}$ during Experiment 1

Figure A.6 represents the evolution over time of the fraction of busy chutes (averaged across replications) for all three policies considered, while Figure A.7 represents an average of their release rates over the same time period. Observe first (from the period prior to the disruption shown in Figure A.6) that while by design all policies have the same risk, ADP operates much closer to gridlock than CWP^{β} and CST^{β} do. The initial average proportion of busy chutes seen there for ADP is around 85%, while CWP^{β} is around 81% and CST^{β} around 75%.

When the disruption occurs, the proportion of busy chutes suddenly increases by about 6% for ADP, 10% for CST^{β} and more than 12% for CWP^{β}. The upper bound of the two standard deviations simulated range for busy chutes goes close to 100% during the experiment for all three policies.

Policy ADP responds to that disruption by quickly decreasing its release rate for a short period of time, which stabilizes within two hours the number of busy chutes to a higher value than normal, but relatively safe nonetheless. The system thus spends little time close

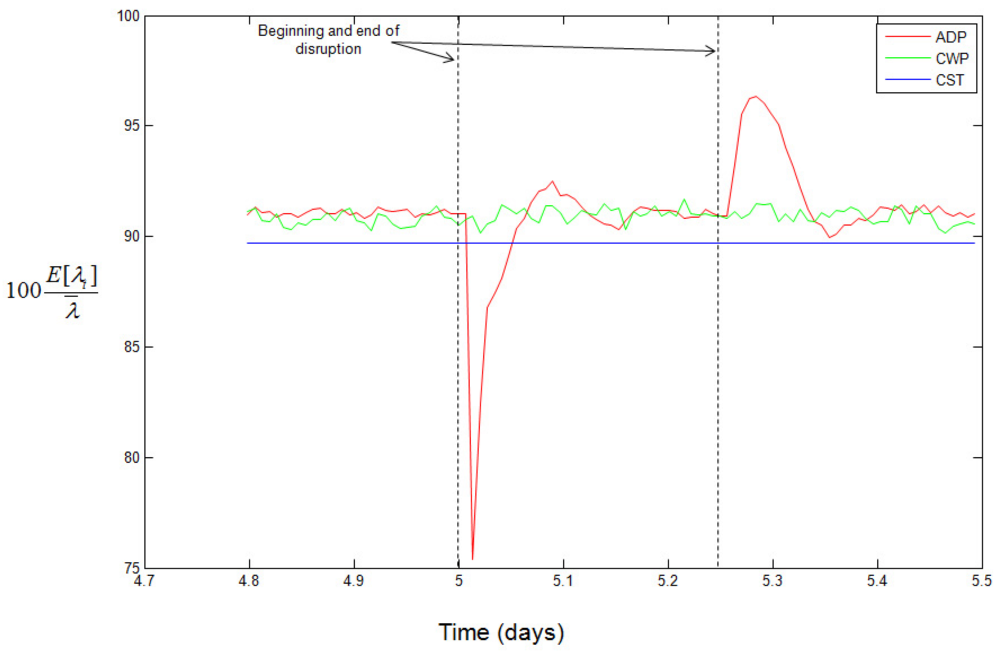


Figure A.7: Evolution of the Average Relative Release Rate $\frac{E[\lambda_t]}{\bar{\lambda}}$ during Experiment 1

to gridlock, which it only experienced in 2.6% of all simulated replications (see Table A.1). As the disruption ends, the second queue is suddenly starved, which ADP sees as an opportunity to release more orders (as seen in Figure A.7), before quickly returning the system to its original steady-state.

The transient disruption considered involves an increase of the transition rate of orders between the first and second queues. As result, it creates a simultaneous decrease of the number X_t in the first queue and an increase of the number Y_t in the second, which leaves the total number in system $X_t + Y_t + Z_t$ relatively unaffected. Consequently, the response to that disruption by policy CWP^β is very muted (if observable at all), as seen in Figure A.7. Because CWP^β adjusts its release rate dynamically to keep $X_t + Y_t + Z_t$ at a constant value, but at the same time the sejourn time of orders in the first queue has decreased, the steady state to which the system converges following the immediate transient response to the disruption is one where the number of busy chutes $Y_t + Z_t$ is maintained at a higher value than before (see Figure A.6). For this reason, CWP is the most dangerous policy in that experiment (under CWP^β the system entered gridlock in 49% of replications, under CWP^γ in all of them). Likewise, the release rate of CST^β remains (by definition) exactly identical throughout the disruption. The initial transient system response under CST^{β} is thus similar to that under CWP^{β}, because for different reasons both policies ignore the initial decrease of X_t and increase of Y_t. However, under CST^{β} the transition rate between the first and second queue starts converging back to the external release rate after an hour or so, causing the number of busy chutes to start decreasing towards its prior steady-state value. For that reason policy CST^{β} fares much better in that experiment than CWP^{β} despite its simplicity, only running the system into gridlock in 5.7% of replications (54% for CST^{γ}, see Table A.1).

A.5.2. Experiment 2: Pick Zone Shutdown Our second experiment consists of temporarily increasing all chute-dwell times (second station service times) by 50% during one hour of simulated time. Is is motivated by the possibility that the incoming flow to the sorter of items originating from a specific pick zone may be temporarily reduced or halted in the actual system – according to our personal communications with managers at our industrial partner, this could be caused for example by a worker omitting to close the pass-through gate of a conveyor belt carrying items from that zone, or an unscheduled interruption of work by pickers in that specific area of the warehouse. As a result, many chutes may remain incomplete until the flow gets back to normal, and the overall throughput of the second queue would decrease. The number of busy chutes would therefore increase, potentially leading to gridlock.

The response of the system to that disruption is best understood by first considering policy CST, because under that policy the input rate to the second queue remains unchanged throughout, so that the evolution of busy chutes over time shown in Figure A.8 is entirely explained by changes in the output rate of the third (packing) queue. Specifically, when the disruption begins the input rate to that queue is suddenly reduced as the service time of all orders in the second queue increases. After a short lag during which packers maintain the overall output by exhausting the queue of orders at the third station, packers are progressively starved and the number of busy chutes therefore quickly increases. As the second queue starts to return towards an equilibrium with a higher number of chutes and its output rate starts to increase back to its original value, packer utilization starts to increase again and the rate at which the number of busy chutes increases starts to drop (this is noticeable in the last third of the disruption period in Figure A.8). Finally, the convergence of the system back to

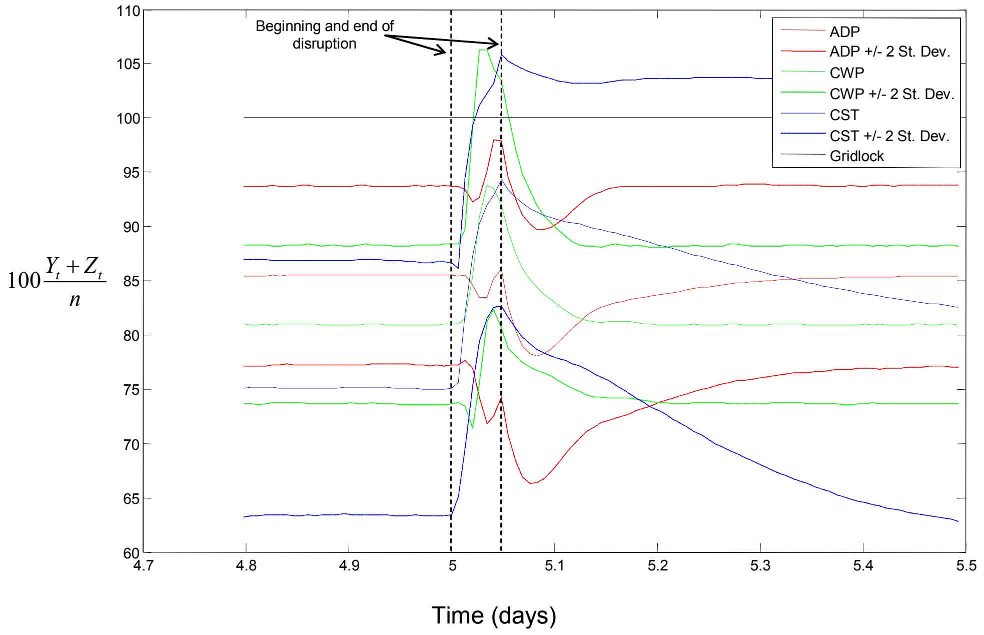


Figure A.8: Evolution of the Average Fraction of Busy Chutes $\frac{Y_t+Z_t}{n}$ during Experiment 2

steady-state is particularly slow under CST, and so is therefore the decrease of the number of busy chutes after the end of the disruption. As a result, the system lingers for a long time in an operating regime that is dangerously close to gridlock (see upper bound of simulated range in Figure A.8), and CST performs worst overall in that experiment among all policies considered (as seen in Table A.1 CST^{β} experienced gridlock in 47.1% of replications, CST^{γ} in 94%).

As seen in Figure A.9 policy CWP does respond to that disruption by sharply decreasing its release rate, however the initial system response under that policy is similar to that under CST (see Figure A.8). This is because CWP's response comes after a lag of about a quarter of the disruption period. We believe that lag to result from several factors; the first is that the initial queue of orders in the (third) packing station is larger than that of CST by about 40%, as the values of $\frac{\mathbb{E}[Z_\infty]}{n}$ corresponding to CST and CWP obtained through simulation for $w = p$ (9.8% and 13.9% respectively) indicate. Depleting that larger queue of work thus allows packers under CWP to slightly postpone starvation relative to CST (as well as the corresponding decrease of packing rate and increase of busy chutes), as described above. Secondly, because CST is only sensitive to changes in the total number of orders in process

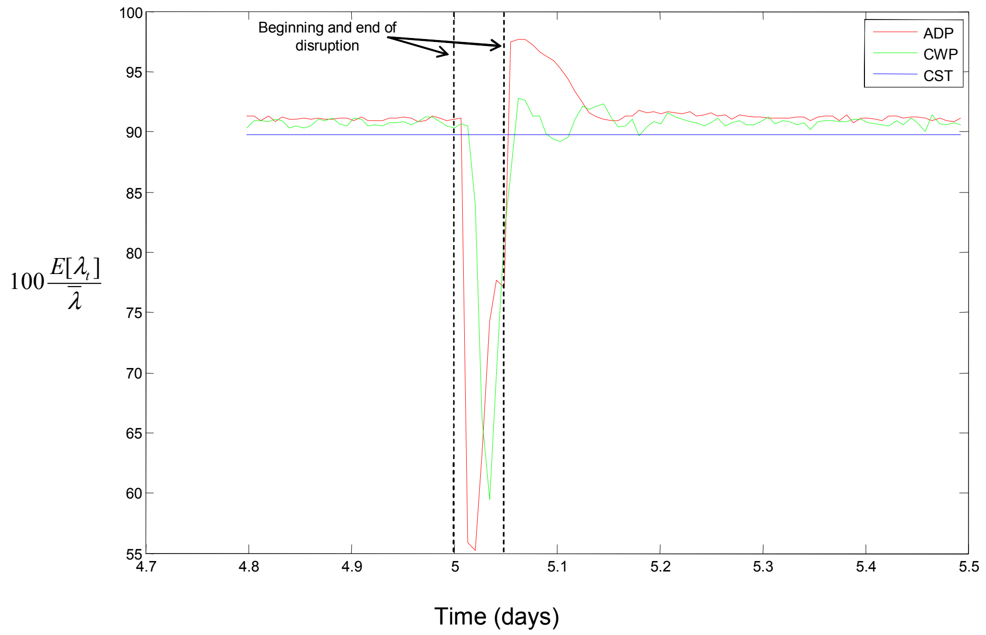


Figure A.9: Evolution of the Average Relative Release Rate $\frac{E[\lambda_t]}{\bar{\lambda}}$ during Experiment 2

$X_t+Y_t+Z_t$, its response is both delayed and muted by the fact that the disruption considered initially changes the number of incomplete chutes Y_t and number of complete chutes Z_t with opposite rates at first, so that their sum remains initially constant. Finally, as all other release control policies in this system CWP may only affect the number of busy chutes after a lag corresponding to the service time in the first station. Overall, CWP only achieves to stop the increase in busy chutes after about two thirds of the disruption period; by then the upper bound of the simulated range for $Y_t + Z_t$ is well above the gridlock level, explaining that the overall performance of CWP in that experiment is only marginally better than that of CST (as seen in Table A.1 CWP^β entered gridlock in 39% of replications, CWP^γ in 68%).

As seen in Figure A.9, the response by policy ADP to that disruption is qualitatively similar to that of CWP, however ADP responds sooner and with a more drastically reduction of its release rate. This is because ADP is sensitive to the individual value of the number of incomplete chutes Y_t, which immediately starts to increase when the disruption begins. Since the initial queue at the third station is longer under ADP (in simulations $\frac{\mathbb{E}[Z_\infty]}{n} = 19.2\%$ for $w = p$), so is the initial period until the depletion of that queue during which $\frac{dZ_t}{dt} \approx -\frac{dY_t}{dt}$ and the number of busy chutes remains approximately constant. These features enable policy ADP to overcome the control lag introduced by the service time at the first station (time-to-chute): Figure A.8 shows that the average number of busy chutes under that policy actually decreases after the initial period just described, before the increase in release rate starting around the middle of the disruption period causes it to increase back to about its original value by the end. Because the variability of the number of busy chutes is increased by that disruption, the upper bound of the simulated range for $Y_t + Z_t$ actually increases during the disruption period, and the system under policy ADP did experience gridlock in 0.5% of replications (see Table A.1). That performance is nevertheless substantially better than that of all other policies considered. Also noteworthy is the behavior of ADP after the disruption ends. As Figure A.8 shows, as the disruption ends and the chute-dwell time (service time of the second station) suddenly increases back to its original value, the average number of busy chutes under both CWP and ADP starts to quickly decreases, whereas that reduction and the return to steady-state are considerably slower under CST. However, Figure A.9 shows that, in contrast to CWP, policy ADP is able to exploit that temporary reduction of the number of busy chutes by temporarily increasing its release rate above its original value, that is push more flow into the system (all while maintaining it in a much safer operating regime, as evidenced by Table A.1).

A.5.3. Experiment 3: Downstream Choke Our last transient experiment consists of temporarily reducing the number of staffed packers by 50% for 10 minutes of simulated time. It is motivated by the possibility in the real system that the operations downstream of the sorter (labelling and shipping) may also experience some disruptions, leading to a limitation of the sorter output due to congestion propagating backwards. This experimental design also constitutes a plausible representation of other types of real system disruptions such as unscheduled breaks by the packers, or a stockout of the empty cardboard boxes available to them. Because this event affects the last queue which is farthest from the admission control point, and because in the scenario considered prior to the disruption ($w = p$) packing already constitutes a bottleneck (see section §5 in the paper), that disruption turns out to be quite severe.

Indeed, Figure A.10 shows that under all policies considered the average number of busy

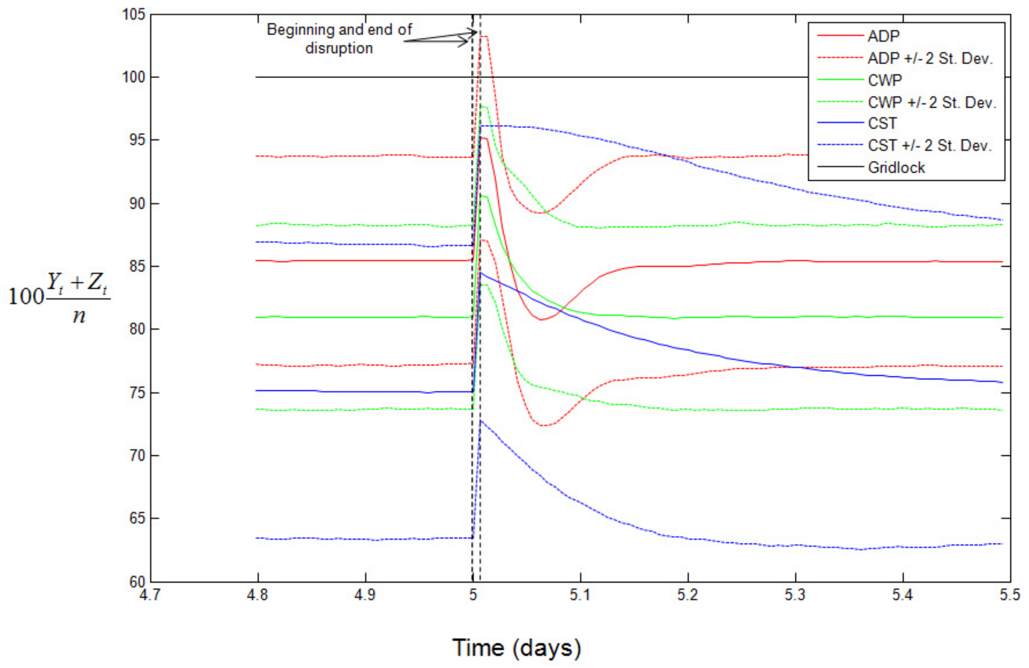


Figure A.10: Evolution of the Average Fraction of Busy Chutes $\frac{Y_t+Z_t}{n}$ during Experiment 3

chutes suddenly and drastically increases as soon as the disruption starts, even though policies CWP^{β} and ADP decrease their respective release rates right away. This is because the control lag introduced by the first station service time is of the same order of magnitude as the disruption period length. As a result, the input rate to the second queue is unchanged for most of the disruption period, while the transient shock considered consists of a sudden reduction of the packing rate (rate of output from the third queue). The policies considered can therefore do little to prevent the increase in busy chutes resulting from the differential between these rates during the disruption period. For this reason, we suggest that the empirical gridlock probabilities for policies ADP, CWP^{β} and CST^{β} reported in Table A.1 (17%, 6% and 4% respectively) reflect more the differences between initial (steady-state) average numbers of busy chutes for these policies before the disruption begins (85.5%, 80.4% and 74.3% respectively, as seen in Table 1 of the paper) than any intrisic differences in how these policies are able to mitigate the disruption. In fact, the average number of busy chutes under ADP, CWP^{β} and CST^{β} is seen on Figure A.10 to increase over the disruption period by approximately 9.5% of the total number n available, regardless of the policy considered.

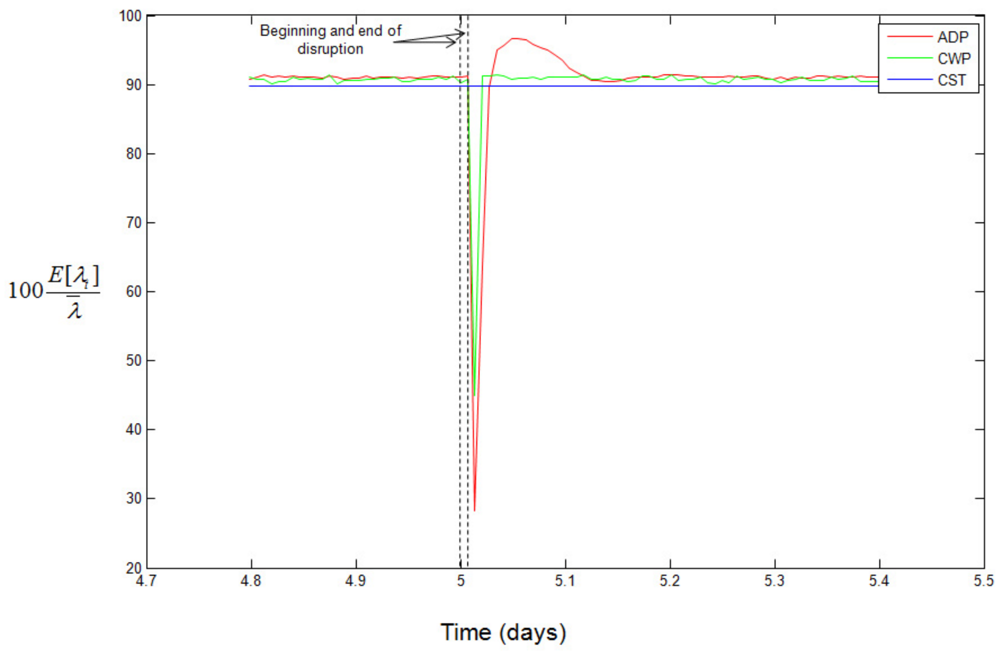


Figure A.11: Evolution of the Average Relative Release Rate $\frac{E[\lambda_t]}{\bar{\lambda}}$ during Experiment 3

From that perspective, the comparison with the policies CWP^γ and CST^γ having the same initial throughput as ADP that is suggested by Table A.1 may be a more grounded one in this setting, and is also favorable to ADP (the system entered gridlock in 42% of replications under CWP^γ, in 81% under CST^γ).

A.6. Item Release Control Study

The sorter capacity depends in part on the number of chutes, the number of staffed packers, and the average chute-dwell time $\mathbb{E}[B]$; the chute-dwell time of each order depends in turn on the number m of items in that order, and the *transit time* T_i necessary for each item $i \in \{1, ..., m\}$ it contains to travel from the pick zone where it is collected to its assigned chute in the sorter. Figure A.12, which displays the timeline of a single customer order with 4 items going through this process, illustrates all the quantities just defined.

An important process feature is that the transit times just defined are highly variable (hence their notation T_i suggesting their modeling as random variables), as they are affected in practice by many factors including: (i) conveyor belt distances between each pickzone and the sorter; (ii) tote congestion encountered on the conveyor system; (iii) time spent on the

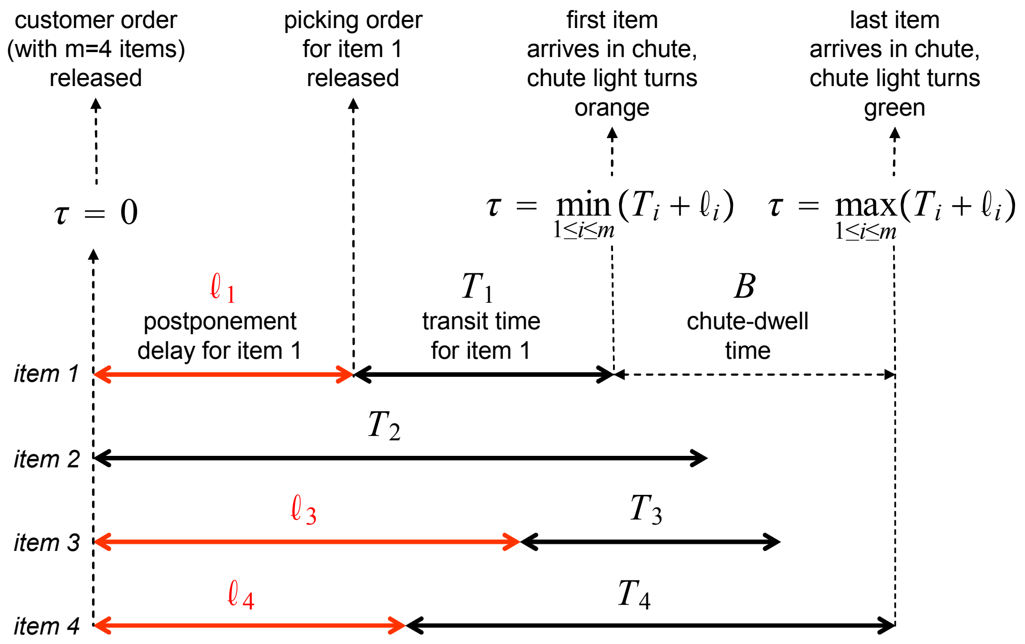

customer order (with m=4 items) released
picking order for item 1 released
first item arrives in chute, chute light turns orange
last item arrives in chute, chute light turns green
$\tau = 0$
$\tau = \min_{1 \le i \le m} (T_i + \ell_i)$
$\tau = \max_{1 \le i \le m} (T_i + \ell_i)$
ℓ_1
postponement delay for item 1
T_1
transit time for item 1
B
chute-dwell time
item 1
T_2
item 2
ℓ_3
T_3
item 3
ℓ_4
T_4
item 4

Figure A.12: Timeline of Customer Order through the Pick-to-Ship Process

circulating buffer; (iv) conveyor system breakdowns; etc. For examples of empirical transit time distributions constructed from historical order flow data in our partner's warehouse, see §A.2. A key observation however is that some of the variability affecting the transit times of items belonging to the same order, for example that resulting from factor (i) above, may be predictable upfront. For instance, consider a customer order for two items (i, j) stored in locations that are very far apart from each other in the warehouse, one being in particular much closer to the sorter than the other, so that $\mathbb{E}[T_i] \ll \mathbb{E}[T_j]$. In this case, it would seem sensible to try and delay the picking of item i at the outset by some delay ℓ_i, so that the two items arrive to the sorter close together and tie up as little chute capacity as possible; one could for example set $\ell_i = \mathbb{E}[T_j] - \mathbb{E}[T_i]$ so that $\mathbb{E}[T_i + \ell_i] = \mathbb{E}[T_j]$. More generally, the problem of item release control consists of setting appropriate postponement lead-times $\ell_i \geq 0$ to delay the picking of each item i of an order, with the goal of reducing its chute-dwell time (see Figure A.12 for an illustration). In the setting of our partner's warehouses, these postponement delays are implemented by slightly altering the waveless picking logic under which orders are released into the active picking assignment queue. Specifically, whenever a customer order is transferred from the first virtual queue of incoming orders to the second picking queue representing the active picking assignments (see section §1 of the paper), the corresponding transfer of any individual items in that order with a positive postponement lead-time (as specified by the control rule just discussed) is then delayed accordingly.

The item release control problem just stated is also explicitly mentioned in Forger (2005), Owyong and Yih (2006), Trebilcock (2007) and, more implicitly, in Le-Duc and de Koster (2005); this type of postponement control mechanism is also identical to that studied in Gallien and Wein (2001). It can be formally stated in the present setting as:

$$\begin{array}{ll} \underset{\ell_1,\ldots,\ell_m}{\textsc{Min}} & \mathbb{E}[B] = \mathbb{E}[\max\limits_{1\leq i\leq m}(T_i+\ell_i) - \min\limits_{1\leq i\leq m}(T_i+\ell_i)] \\ \text{subject to:} & \ell_i \geq 0\ \forall i \end{array}, \tag{A.5}$$

where $T_1,\ldots,T_m$ are random transit time variables with specified distributions. Note that the notation $\mathbb{E}[B]$ used in (A.5) is not coincidental, as this objective of this optimization problem is exactly equal to the expected value of the chute-dwell time, as defined in section §3.1 of the paper. Likewise, the *time-to-chute* or service time of the first queueing station discussed in that section can be expressed for each order in terms of the transit time variables as $A = \min\limits_{1\leq i\leq m}(T_i+\ell_i)$.

The remainder of this section presents some work performed with the goal of identifying an item release control policy improving upon the one used by our partner. It contains the statement of theoretical results obtained under specific distributional assumptions for the transit times (in §A.6.1), an implementation study (in §A.6.2) and an impact assessment study (in §A.6.3).

A.6.1. Analysis. We first state formally our optimality result for the optimization problem (A.5) just stated.

Proposition 1 *Let $m \in \mathbb{N}\backslash\{0,1\}$, $k > 0$ and $(T_1,\ldots,T_m)$ be m independent r.v.'s such that for all i, $T_i \sim CMT1^k(\alpha_i)$ for some $\alpha_i > 0$ (or equivalently $\mathbb{P}(T_i \leq \tau) = \exp(-\alpha_i e^{-k\tau})$), then the vector $(\ell_1^*,\ldots,\ell_m^*)$ defined by*

$$\ell_i^* \triangleq \left(\max_{j\in\{1,\ldots,m\}} \mathbb{E}[T_j]\right) - \mathbb{E}[T_i] \tag{A.6}$$

satisfies $\min\limits_{i\in\{1,\ldots,m\}} \ell_i^* = 0$ and is an optimal solution to (A.5).

Proof: Note that if $\mathbb{P}(T_i \leq \tau) = \exp(-\alpha_i e^{-k\tau})$ then $\mathbb{E}[T_i] = \frac{\gamma+\ln\alpha_i}{k}$ where γ denotes Euler constant, and since the family of r.v's $(T_1,\ldots,T_m)$ is closed under maximization and translation we have

$$\mathbb{E}[\max(T_1+\ell_1,...,T_m+\ell_m)] = \frac{\gamma + \ln\left(\sum_{i=1}^m \alpha_i e^{k\ell_i}\right)}{k}. \qquad (A.7)$$

Also

$$\begin{aligned}\mathbb{P}(\min(T_1,...,T_m) > \tau) &= \prod_{i=1}^m (1-\exp(-\alpha_i e^{-k\tau}))\\ &= 1 + \sum_{i=1}^m (-1)^i \sum_{\{\beta_1,...,\beta_i\}\subset\{\alpha_1,...,\alpha_m\}} \exp\left(-\left(\sum_{j=1}^i \beta_j\right) e^{-k\tau}\right)\end{aligned}$$

so that

$$\mathbb{E}[\min(T_1+\ell_1,...,T_m+\ell_m)] = \sum_{i=1}^m (-1)^{i+1} \sum_{\{\beta_1,...,\beta_i\}\subset\{\alpha_1 e^{k\ell_1},...,\alpha_m e^{k\ell_m}\}} \frac{\gamma + \ln\left(\sum\limits_{j=1}^i \beta_j\right)}{k}. \qquad (A.8)$$

The objective function $f(\ell_1,...,\ell_m)$ of (A.5) can thus be expressed in closed form as the difference of the left-hand sides of (A.7) and (A.8). The first-order optimality condition of the unconstrained problem is then obtained from

$$\frac{\partial f(\ell_1,...,\ell_m)}{\partial \ell_i} = \frac{\alpha_i e^{k\ell_i}}{\sum_{j=1}^m \alpha_j e^{k\ell_j}} - 1 + \sum_{j=1}^{m-1} (-1)^{j+1} \sum_{\{\beta_1,...,\beta_j\}\subset\{\alpha_1 e^{k\ell_1},...,\alpha_m e^{k\ell_m}\}\setminus\{\alpha_i e^{k\ell_i}\}} \frac{\alpha_i e^{k\ell_i}}{\alpha_i e^{k\ell_i} + \sum_{s=1}^j \beta_s}. \qquad (A.9)$$

It can be easily seen through direct substitution in (A.9) that for any constant $\phi > 0$ the solution $(\ell_i^{(\phi)})_{i\in\{1,...,m\}}$ defined by

$$\ell_i^{(\phi)} \triangleq \frac{1}{k}\ln\frac{\phi}{\alpha_i} \text{ or } \alpha_i e^{k\ell_i^{(\phi)}} = \phi \text{ for all } i$$

solves the first-order condition $\frac{\partial f(\ell_1,...,\ell_m)}{\partial \ell_i} = 0$. Note that setting $\phi = \max_{i\in\{1,...,m\}} \alpha_i$ yields $\ell_i^{(\phi)} \geq 0$ for all i, $\min\limits_{i\in\{1,...,m\}} \ell_i^{(\phi)} = 0$ and

$$\begin{aligned}\ell_i^{(\phi)} &= \frac{1}{k}\left(\ln\left(\max_{j\in\{1,...,m\}} \alpha_j\right) - \ln\alpha_i\right)\\ &= \left(\max_{j\in\{1,...,m\}} \mathbb{E}[T_j]\right) - \mathbb{E}[T_i].\end{aligned}$$

To finally prove that a solution of the first-order condition is optimal it suffices to observe that the objective $f(\ell_1,...,\ell_m)$ is convex, since $\mathbb{E}[\max(T_1+\ell_1,...,T_m+\ell_m)]$ is convex in

$(\ell_i)_{i\in\{1,...m\}}$ and $\mathbb{E}[\min(T_1+\ell_1,...,T_m+\ell_m)]$ is concave in $(\ell_i)_{i\in\{1,...m\}}$.

The distributional assumptions stated in Proposition 1 imply that the transit times follow Gumbel distributions with the same variance, a particular example of a distributional family that is closed under maximization and translation (see Gallien and Wein 2001). However, notice that the objective function in (A.5) is a range, thus involving not just the maximum but also the minimum of translated random variables. Because the family of Gumbel distributions just mentioned is not closed under minimization or range, it is noteworthy that problem (A.5) can still be solved in closed form. Observe also that there are infinitely many optimal solution to (A.5), as the range operator is not affected by translations so that $(\ell_i+\tau)_{1\leq i\leq m}$ will be optimal for any $\tau \geq 0$ provided $(\ell_i)_{1\leq i\leq m}$ is. The additional condition $\min_{i\in\{1,...,m\}} \ell_i = 0$ is thus not justified by the formulation of problem (A.5) alone, but rather by context as any solution with a higher value of $\min\limits_{i\in\{1,...,m\}} \ell_i$ would not improve the expected chute-dwell time relative to (ℓ_i^*) defined by (A.6), however it would increase the cycle time of the corresponding customer order.

Finally, note that the optimal solution described in Proposition 1 is also optimal for the deterministic approximation of problem (A.5) obtained by replacing each transit time T_i by a deterministic equivalent $\mathbb{E}[T_i]$. Proposition 1 thus provides a theoretical justification for the intuitive solution proposed for the two-item order example mentioned earlier, and more generally for the deterministic approximation just described. However, we caution the reader that the validity of such approximation is far from general and seems very sensitive to model details, as that same approximation turns out to be quite coarse in other very similar settings (see Gallien and Wein 2001).

Interestingly, our industrial partner had already implemented release postponement delays using an equation very similar to (A.6) before interacting with us, although it had developed that formula from intuition (which we suspect meant applying the deterministic approximation mentioned earlier), as opposed to a formal analysis of the type just presented. More specifically, our partner would measure a few times per year the medians ξ_q of the transit times corresponding to each pick zone $q \in \{1,...,\bar{q}\}$. For each order with m items indexed by $i \in \{1,...,m\}$ received between those measurement updates, it would then determine the set of pick zones $\mathbf{q} \triangleq (q_i)_{i\in\{1,...,m\}}$ from which these items should be picked, and implement postponement delays given by

$$\hat{\ell}_i(\mathbf{q}) = \left(\max_{j \in \{1,...,m\}} \xi_{q_j} \right) - \xi_{q_i}. \tag{A.10}$$

However, that policy seemed to ignore some important drivers of predictable transit times variability. In the following subsection, we describe the study we performed in order to implement the theoretical analysis presented earlier, and attempt to improve upon the item release control policy (A.10) just described.

A.6.2. Implementation Study. Our first goal was to characterize the transit time distributions using all the information available at the time when each order was released, so that the item postponement delays implemented then through equation (A.6) would rely on estimators of the corresponding expected transit times that would be as accurate as possible. We first realized when visiting the facility that each individual pick zone $q \in \{1, ..., \bar{q}\}$, defined by our industrial partner to partition the entire picking area for labor allocation purposes, still covered a relatively large space and sometimes spanned several floors. In particular, two totes originating from the same pick zone could travel towards the sorter on conveyor paths with lengths differing by several hundred meters, so that the existing subdivision of the picking area into pick zones seemed too coarse for our purposes. As a result, we defined a finer partition of the picking area into smaller subdivisions that we called *conveyor zones*, indexed in the following as $r \in \{1, ..., \bar{r}\}$ with $\bar{r} \gg \bar{q}$. These were specifically defined so that the actual lengths of the conveyor paths followed by totes released in the same conveyor zone would be considerably more homogeneous.

In order to better understand additional drivers of predictable variability for the item transit times, we constructed and examined the empirical distributions of transit times for items picked within the same conveyor zone, following the overall methodology described in §A.2. This data analysis work resulted in a set of fitted $CMT1$ transit time distributions $\mathcal{T} \triangleq \{T(r,g) : r \in \{1, ..., \bar{r}\}, g \in \{1, ..., \bar{g}\}\}$, specifically one for each conveyor zone and congestion level. Combining these distributions with the analysis described in §A.6.1, the item release control policy that we recommended implementing for each customer order consisted of:

1. Identifying the current system congestion level g;

2. Identifying the conveyor zone $r_i \in \{1, ..., \bar{r}\}$ in which each item $i \in \{1, ..., m\}$ of the customer order is to be picked;

3. Postponing the release of the picking order for each item $i \in \{1, ..., m\}$ by the lead-time

$$\ell_i^*(\mathbf{r}, g) = \left(\max_{j \in \{1,...,m\}} \mathbb{E}[T(r_j, g)] \right) - \mathbb{E}[T(r_i, g)], \qquad (A.11)$$

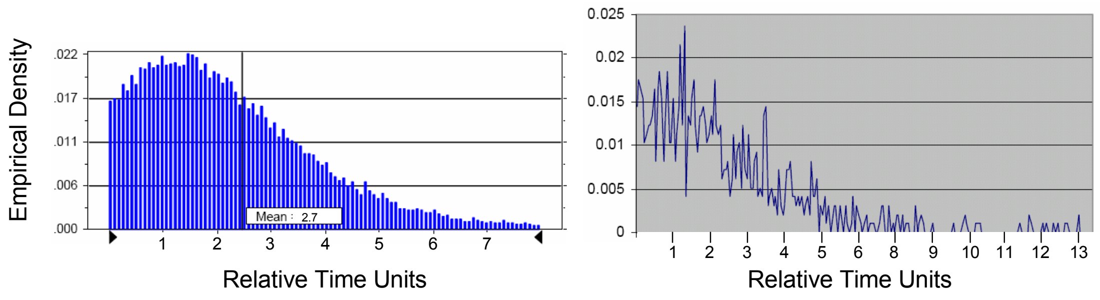
where the dependence on $\mathbf{r} \triangleq (r_i)_{i\in\{1,...,m\}}$ and g is shown explicitly.

Note that even in the theoretical setting where the item transit time distributions for each order would be independent and exactly given by the $CMT1$ distributions from the set $\mathcal{T}$ defined above, the policy defined by equation (A.11) would likely be suboptimal. This is because the fitted $CMT1$ distributions $T(r, g)$ obtained from the data analysis just described do not belong to $CMT1$ families that are closed under maximization and translation. That is, two distributions $T(r, g)$ and $T(r', g)$ from the set $\mathcal{T}$ of fitted distributions corresponding to different conveyor zones r and r' but the same congestion level g do not typically have the same variance. This represents a deviation from the assumptions required for Proposition 1 to hold and the policy defined by (A.11) to be rigorously optimal, although it is not clear what the corresponding optimality loss amounts to. Furthermore, it is also not clear how damaging the independence assumption for these transit times actually is. Motivated by these questions, the following section provides an assessment of the potential benefits to be derived from an implementation of policy (A.11) in our industrial partner's warehouse.

A.6.3. Impact Assessment. Our methodology was to develop a Monte-Carlo simulation model that could provide a reliable prediction of the expected chute-dwell time associated with our proposed release control policy $\ell^* \triangleq (\ell_i^*)_{i\in\{1,...,m\}}$ defined by (A.11), enabling a comparison over a large number of customer orders with that resulting from the release control policy $\hat{\ell} \triangleq \left(\hat{\ell}_i\right)_{i\in\{1,...,m\}}$ used by our industrial partner before our interaction and defined by (A.10). As a first step designed to assess our model's predictive accuracy and establish a baseline for comparison, we simulated the random variable defined for each congestion level $g \in \{1, ..., \bar{g}\}$ as

$$B(\hat{\ell}, g) \triangleq \max_{1\le i\le M} (T(R_i, g) + \hat{\ell}_i(\mathbf{Q})) - \min_{1\le i\le M} (T(R_i, g) + \hat{\ell}_i(\mathbf{Q})), \qquad (A.12)$$

where M (number of items per order) and $\mathbf{R} \triangleq (R_i)_{i\in\{1,...,M\}}$ (congestion zones in which the items are to be picked) were random variables following empirical distributions $\{\mathbb{P}(M = m) : m \geq 2\}$ and $\{\mathbb{P}(R_i = r) : r \in \{1, ..., \bar{r}\}\}$ constructed from the first half of our data set (referred to in the following as the *training set*), and such that for every realization of M, $R_1, ..., R_M$ are i.i.d. The number of components of random vector $\mathbf{Q} \triangleq (Q_i)_{i\in\{1,...,M\}}$ was likewise determined by the outcome of the r.v. M, with each value $Q_i \in \{1, ..., \bar{q}\}$ for $i \in \{1, ..., M\}$ given by the pick zone containing the simulated congestion zone R_i. In order to evaluate the predictive accuracy of the simulation model defined by (A.12), we then compared its output with the empirical distribution of the actual chute-dwell time for each congestion level B_g constructed directly from the second half of our data set (the *evaluation set*). For illustration purposes, Figures A.13 (a) and (b) contain plots of the empirical probability density functions (p.d.f.) of $B(\hat{\ell}, 6)$ and B_6, obtained respectively by Monte-Carlo simulation and direct construction from the evaluation set, for congestion level $g = 6$. Note that the effective support of these distribution starts at zero, reflecting cases where all items from the same order travel through the process in the same tote, a typical occurence for orders consisting of several identical items.



(a) Simulated Chute-Dwell Time $B(\hat{\ell}, 6)$, data from training set

(b) Actual Chute-Dwell Time B_6, measured from evaluation set

Figure A.13: Empirical Density of Actual and Simulated Chute-Dwell Time for Congestion Level $g = 6$

More generally, the predicted (resp. measured) expected chute-dwell times $\mathbb{E}[B(\hat{\ell}, g)]$ (resp. $\mathbb{E}[B_g]$) obtained by applying the method just described for all congestion levels are shown in the second and third columns of Table A.2, with the relative prediction error $\left(\mathbb{E}[B(\hat{\ell}, g)] - \mathbb{E}[B_g]\right) / \mathbb{E}[B_g]$ shown in its fourth column. Using the relative picking frequencies of each congestion level in our entire data set to weight these relative prediction errors, we computed an estimate of the overall relative prediction error associated with our simula-

Congestion Level g	Simulated Chute-Dwell Time $\mathbb{E}[B(\hat{\ell},g)]$	Actual Chute-Dwell Time Data $\mathbb{E}[B_g]$	Relative Prediction Error $\frac{\mathbb{E}[B(\hat{\ell},g)]-\mathbb{E}[B_g]}{\mathbb{E}[B_g]}$	Optimized Chute-Dwell Time $\mathbb{E}[B(\ell^*,g)]$	Relative Improvement Potential $\frac{\mathbb{E}[B(\ell^*,g)]-\mathbb{E}[B(\hat{\ell},g)]}{\mathbb{E}[B(\hat{\ell},g)]}$
1	2.779	2.784	-0.15%	2.739	-1.44%
2	2.610	2.628	-0.68%	2.562	-1.84%
3	2.298	2.280	0.79%	2.274	-1.04%
4	2.904	2.904	0%	2.862	-1.45%
5	2.736	2.748	-0.43%	2.730	-0.22%
6	2.604	2.574	1.17%	2.574	-1.15%
7	2.724	2.706	0.67%	2.694	-1.10%

Table A.2: Simulation Results for Item Release Control Experiments

tion model as 0.53%, a result establishing in our view the relatively high predictive power of our simulation model.

As a second step designed to assess the relative improvement potential of our suggested policy ℓ^*, we simulated for each congestion level $g \in \{1, ..., \bar{g}\}$ the random variable

$$B(\ell^*, g) \triangleq \max_{1\leq i\leq M}(T(R_i, g) + \ell_i^*(\mathbf{R}, g)) - \min_{1\leq i\leq M}(T(R_i, g) + \ell_i^*(\mathbf{R}, g)), \tag{A.13}$$

where the random variables M and $\mathbf{R} \triangleq (R_i)_{i\in\{1,\dots,M\}}$ were generated as in (A.12), and the item release postponement delays $\ell_i^*(\mathbf{R}, g)$ were computed for each random order realization according to (A.11). The fifth and sixth columns of Table A.2 respectively show the estimated average chute dwell time $\mathbb{E}[B(\ell^*, g)]$ obtained for each congestion level through the Monte-Carlo simulation procedure just described, and the corresponding predicted relative improvement potential $\left(\mathbb{E}[B(\ell^*, g)] - \mathbb{E}[B(\hat{\ell}, g)]\right)/\mathbb{E}[B(\hat{\ell}, g)]$.

We believe that the bias introduced by the relative prediction error of our simulation model shown in the fourth column of Table A.2 for each congestion level should equally apply to both the model's prediction for the current policy $\hat{\ell}$ and that for our proposed policy ℓ^*; this is because that bias most likely follows from our estimation of the transit time distributions, as opposed to the specific values of the postponement delays. Accordingly, the results shown in the last column of Table A.2 suggest that our proposed policy ℓ^* should in practice decrease the average chute-dwell time by at least 1% in six congestion levels out of seven, and by 0.22% in the last one. The data available to us however does not allow us to further back up the claim that the bias of our simulation model relative to the real system applies to both $\hat{\ell}$ and ℓ^* in a systematic way. One could thus also consider

Number of Packers w	p	$112.5\%p$	$125\%p$
$100 \times \frac{\gamma_*^{ADP} - \gamma^{ADP}}{\gamma^{ADP}}$	0	0.27	0.29

Table A.3: Relative Impact on Throughput of a 1.18% Reduction in Chute-Dwell Time

a more pessimistic (in our view overly conservative) scenario where our simulation model would in fact overestimate the true performance of our proposed policy ℓ^* whenever that model underestimates the performance of the current policy $\hat{\ell}$ (that is, whenever the relative prediction error in Table A.2 is positive). Assuming that relative overestimation for ℓ^* to have the same absolute value than the corresponding underestimation for $\hat{\ell}$, our improvement prediction results would still be significant in all congestion zones but one, with 3 of them having a predicted average chute-dwell time reduction larger than 1.44%, and the other three having predicted reductions of 0.4%, 0.25% and 0.22% respectively.

In addition, the queueing model discussed in section §3.1 of the paper, because it includes chute-dwell time as primary input data, is also useful for evaluating the impact of the item release policy we developed on throughput (which arguably constitutes a more relevant performance measure than average chute-dwell time). Specifically, we used that model to simulate the impact on throughput of the 1.18% average reduction of chute-dwell time predicted in §A.6.3, with the policy ADP obtained with risk level $\underline{\beta}$ and under various staffing scenarios (as defined in section §5.1 in the paper). The results are shown in Table A.3[5], where γ_*^{ADP} refer to the average throughput obtained with the reduced chute-dwell times, while γ^{ADP} denotes the average throughput obtained with the original chute-dwell times used in our earlier experiments.

Consistent with the results discussed in Section §5.2 of the paper, Table A.3 shows that in the scenario with the same number of packers as used by our partner (p), packing is effectively the relevant system-wide constraint on throughput, so that improving other parts of the system (such as chute-dwell time) has little impact if any. Even with more packers however, the predicted impact of the chute-dwell time reduction attributable to our proposed item release policy remains very small. While disappointing in a way, these results thus provided the useful determination that the potential for improving our partner's warehouse

[5] Table notes: the estimates of throughput γ_*^{ADP} obtained have a standard estimation error from simulation lower than 0.04%.

operation through better item release control policies was most likely very limited.

References

Bertsekas, D. P. and J. N. Tsitsiklis (1996). *Neuro-Dynamic Programming*. Belmont, Massachusetts: Athena Scientific.

Forger, G. (2005, January). Joltin' java DC. *Modern Materials Handling*.

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.

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.

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.

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