Order Release Control for an Online Retailing Warehouse

Jérémie Gallien[1] and Théophane Weber[2]

September 14, 2007

Abstract

Working in collaboration with a large online retailer, our goal is to develop an operational solution to the problem of order release control for its highest volume and most automated warehouse pick-to-ship process. This problem consists of dynamically varying the rate at which new picking orders are released into this process in order to achieve a high throughput while mitigating the risk of congestion-induced collapse (gridlock). We describe a queueing model of this complex process with validated predictive accuracy against actual historical data, and develop a numerical approximate solution method for an associated constrained dynamic program, which we implemented to compute our proposed policy. Simulation experiments suggest that an implementation of our computed policy along with an increase in staffed packers by 25% could increase process throughput by 10%. They also shed light on why our proposed policy outperforms other simple release policies such as constant release rate and CONWIP in this setting, in terms of both steady-state throughput and robustness to transient disruptions.

1. Introduction

Order fulfillment is a critical challenge for online retailers (*e-tailers*). In particular, warehouse outbound or *pick-to-ship* operations directly impact customer satisfaction and, because they are increasingly automated, account for a large fraction of capital expenses. Depending on the volume, weight and number of items ordered by customers, e-tailers may use different processes for these pick-to-ship operations. The work to be described here has been conducted in collaboration with a large e-tailer (*our industrial partner*) with the goal of identifying better control policies for one such process, specifically that used for customer orders containing multiple items that are all small and light enough to be both transportable on conveyors and processed by an automatic sorting machine (*sorter*). This particular process is significant because it supports a large fraction of customer orders, and is especially capital-intensive and complex – it is subject in particular to congestion-induced collapse, that is a rapid deterioration of its throughput when congestion becomes excessive. It also constitutes an example of how mass customization can be achieved in a warehouse setting.

[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

More specifically, this paper focuses on the problem of *order release control*, which was perceived by our industrial partner when we started interacting as an important area of improvement opportunity. That is, our main goal is to derive a control policy for dynamically adjusting the release rate of picking orders over time so as to maximize process throughput while keeping the probability of congestion-induced collapse under an acceptable threshold.

The remainder of this paper is structured as follows: after a more detailed process description in §1.1, we provide a more detailed discussion of the problem considered in §1.2 then a review of the related literature in §1.3. Our model and analysis methods are described in §2, our numerical experiments and results are discussed in §3, and we offer concluding remarks in §4. The Appendix contains supporting material, including auxiliary results and detailed algorithm statements. Throughout this paper mathematical variables in capital letters refer 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.

1.1. Process Description. The process we focus on in this paper is represented pictorially in Figure 1, and starts with the release of customer orders into the work input queue of the warehouse's picking area.

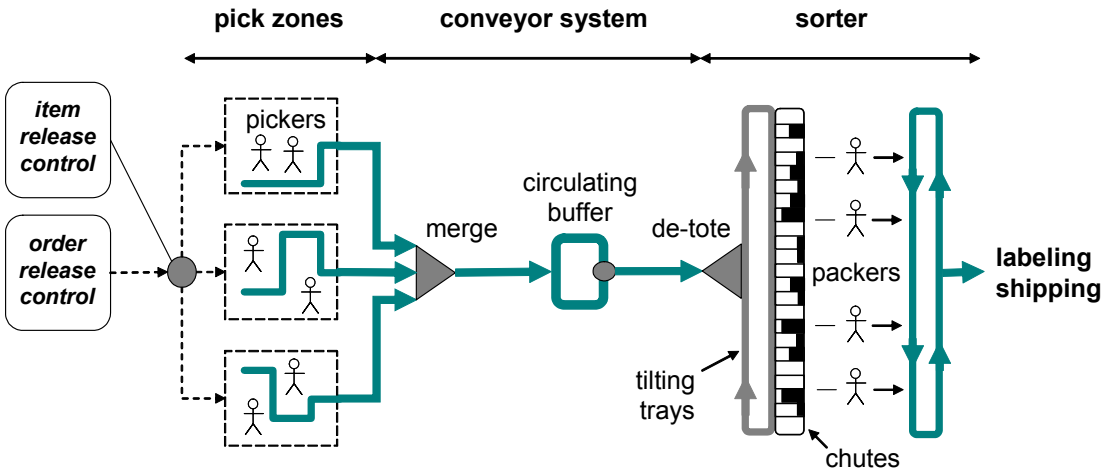


Figure 1: Flow Diagram of the Pick-to-Ship Process Considered

Because of the very large space required to store all items held in inventory at the ware-

house, the picking area is subdivided into several zones (*pick zones*) having different locations within the warehouse, each having its dedicated team of workers (*pickers*). For this reason, a decision rule referred to as *item release control* applies immediately after each customer order release, and prescribes the timing for the release of the picking instructions for the individual items in that order into the appropriate pick zones (see below). From that point, pickers collect items ordered by customers onto individual rolling carts carrying several plastic containers (*totes*) that are offloaded when full onto a conveyor belt spreading through their pick zone (the number of items fitting in each tote varies according to individual item packing volumes). Note that different items from the same customer order are thus likely to be collected by different pickers in different pick zones. Conveyor belts carrying totes coming out of all the pick zones then merge and lead first to a circulating buffer where selected totes may be temporarily held, then finally to the automated sorter where all items from the same customer order are reunited before being packed together and shipped.

More specifically, the sorter consists of a large number of chutes that are each temporarily assigned to a customer order. Individual items are taken from totes coming from the circulating buffer and placed on trays being carried in front of all the chutes (*de-tote*). Each tray then tilts as it arrives in front of the appropriate chute, dropping then the item it carries so it joins there any other items from the same order that may already be present. An orange light attached to each chute turns on when it receives the first item from the customer order it has been temporarily assigned to, and that light turns from orange to green when the last item of an order arrives, signalling to the workers assigned to the sorter (*packers*) that this order is complete and these items are ready to be packed. The packing operation consists of manually placing all items from the chute into a cardboard shipping box, which is then sent to the labelling and shipping departments through a secondary conveyor belt; this turns off the light attached to the chute, which becomes then available for another customer order. Note that each sorter chute thus continuously cycles through three states in the following order: (i) empty/unassigned/lights off; (ii) incomplete/waiting for items/orange; and (iii) complete/waiting for a packer/green. The sorter capacity thus depends in part on the number of chutes, the number of staffed packers, and the average time $\mathbb{E}[B]$ between the arrival of the first and last items of an order in every chute (*chute-dwell time*); this in turn depends on the number m of items in that order, and the time T_i necessary for each item $i \in \{1, ..., m\}$

it contains to travel from the work input queue of the pick zone where it is collected to its assigned chute in the sorter (*transit time*). Figure 2, which displays the timeline of a single customer order with 4 items going through this process, illustrates all the quantities just defined.

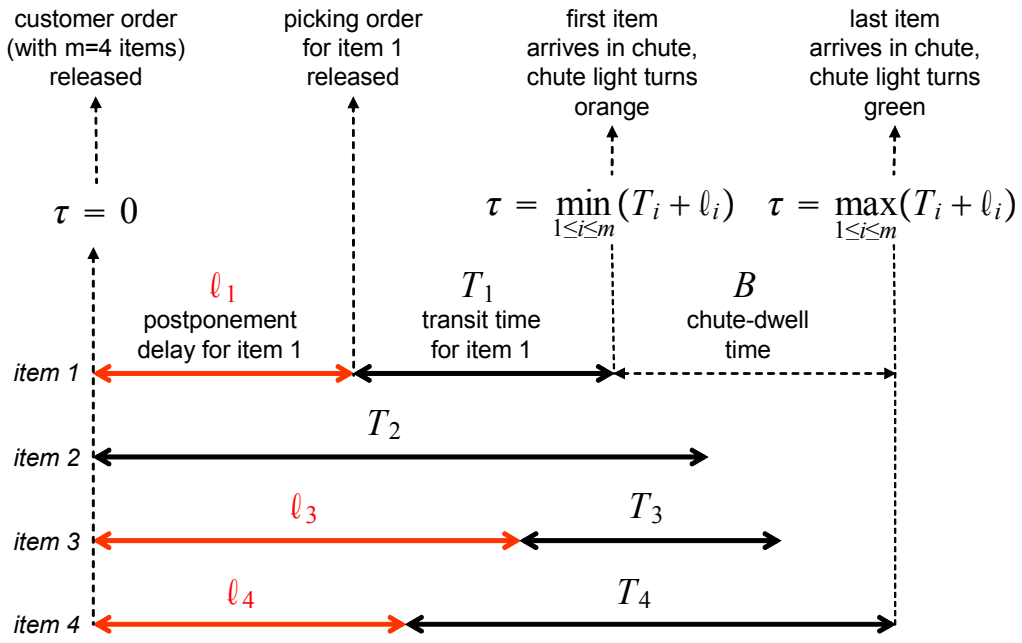


Figure 2: Timeline of Customer Order through the Pick-to-Ship Process

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 circulating buffer; (iv) conveyor system breakdowns; etc. However, a key observation 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 postpone the release of the picking order for 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 mentioned above consists of setting appropriate postponement lead-time $\ell_i \geq 0$ to delay the picking of each item i it contains, at the time when this order as a whole is released (see Figure 2). This type of control mechanism is effectively identical to that studied in Gallien and Wein (2001), and one of our first contributions as part of that collaboration was in fact to formulate and solve (under some distributional assumptions) the optimization problem

$$\begin{array}{ll} \underset{\ell_1,\ldots,\ell_m}{\text{MIN}} & \mathbb{E}[B] = \mathbb{E}[\max_{1\leq i\leq m}(T_i+\ell_i) - \min_{1\leq i\leq m}(T_i+\ell_i)] \\ \text{subject to:} & \ell_i \geq 0 \ \forall i \end{array}, \tag{1}$$

with the hope of identifying an item release control policy improving upon the one used by our partner. While we refer the reader to the Appendix for a more detailed discussion and mathematical proof, it turns out that the optimal solution

$$\ell_i^* \triangleq \left(\max_{j\in\{1,\ldots,m\}} \mathbb{E}[T_j]\right) - \mathbb{E}[T_i] \tag{2}$$

to problem (1) that we derived was in fact very similar to that already in place, although it had been identified by our partner through intuition as opposed to mathematical analysis. In any case our efforts in this area, which are described in the Appendix, resulted in the presumably useful determination that the potential for increasing process throughput through better item release control was fairly limited.

Finally, the data describing the flow of orders through this pick-to-ship process was readily available. Specifically, our industrial partner continuously records in a database the times of all the events affecting the flow of every individual item, tote and customer order going through this process, including release for picking, picking, tote entry into circulating buffer, item arrival in sorter chute, packing start, packing end, and many more. This feature turned out to be a key enabler of our analysis.

1.2. Problem Background. The two primary control levers routinely available to managers for dynamically affecting process congestion and throughput in the pick-to-pack process just described include order release control and staffing. These two levers are obviously related, and in fact some of the results we report in §3 shed some light on the impact of staffing decisions for packers on throughput. However, because the time frequency at which order release control decisions are made is significantly higher than that

corresponding to staffing decisions, it seems legitimate to focus on release control assuming that staffing decisions constitute a fixed exogenous parameter, as we do here. Release control turns out to be a challenging lever to operate in this setting however. On the one hand, the intense pressure to achieve high throughput levels and run the sorter at capacity because of stringent customer lead-time commitments and financial considerations creates a natural tendency to release as many picking orders as available. On the other hand, the particular dynamics of this process also creates a heavy penalty for excessive work-in-process inventory. 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 starts to build up. As a result, the very items needed to complete orders tying up chutes and relieve this congestion can no longer reach the sorter because of... the same congestion. In such conditions, the overall process throughput quickly deteriorates and the sorter comes to a standstill, a state fittingly referred to as *gridlock*. The only remedy is then to partly stop the conveyor system and manually transport items stuck upstream to their matching sorter chutes, a lengthy and laborious procedure entailing a considerable loss of production. In summary, the pick-to-ship process considered is prone to congestion-induced collapse – see Alderson (2005) for relevant references and examples of other systems exhibiting this behavior.

At the outset of our interaction, managers at our industrial partner could in practice affect this order release rate throughout the day in at least two ways: (i) by changing a software setting for the maximum number of customer orders allowed in the pick zones, which may be done at a relatively high frequency if desired (e.g. a few times per hour); and (ii) by changing the number of pickers on staff, which for practical reasons may only be done at a much lower frequency (e.g. a few times per day). While the relationship between these two controls and the resulting outgoing rate of customer orders onto the conveyor system was fairly well understood (although we are not at liberty to discuss it further here), no formal guidelines had been developed for dynamically setting that release rate as a function of observed process conditions (conveyor system congestion, numbers of complete, incomplete and unassigned sorter chutes) and packers' staffing level (see §1.1). 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 had been established empirically and were not supported by any analysis, appeared inconsistent across managers

and facilities, and adherence to those guidelines was not enforced. Not coincidentally, the occurence of gridlock during peak demand periods was more frequent than desired. Struck by the lack of a conceptual framework for making decisions with such important consequences, our industrial partner and we thus decided to focus our collaboration on developing a quantitative model allowing to generate operational order release guidelines adapted to such peak demand periods.

1.3. Literature Review. As emphasized in Xu (2005), relatively little research work has been done so far on the many operational problems specific to online retailing. Among those, problems related to the management of e-tailers' warehouses (also called *fulfillment centers*) are no exception. Bartholdi and Hackman (2006) provide a useful overview of quantitative models of warehouse operations, including some that are also relevant and useful in an e-tailing setting. However, their survey does not mention models that capture the managerial challenges specific to e-tailing warehouses. On this subject, useful process descriptions and contextual information may be found in the Master's theses of Bragg (2003) on process design, Zeppieri (2004), Lieu (2005) and Rubenstein (2006) on inventory positionning, Jackson (2006) on inbound receiving, and Price (2004) and Faranca (2004) on simulation models of pick-to-ship processes. Allgor, Graves and Xu (2005) describe a two-echelon inventory planning model with space constraints designed to understand some of the trade-offs involved when placing inventory in the reserve and forward picking areas of an e-tailing warehouse (see also the more generic models in Chapter 8 of Bartholdi and Hackman 2006). Xu (2005) also presents a model and heuristics to determine how customer orders should be assigned to one of several fulfillment warehouses belonging to an e-tailer. The present paper focuses on the order release control problem immediately following that warehouse assignment decision. To the best of our knowledge, it is the first to present optimization models designed to generate formal work release control policies for an e-tailing fulfillment center.

The specific warehouse control problem that we consider is related to the relatively large body of existing research work on the control of admissions into queueing systems – see Stidham (1985) for an early survey. The salient features of our model relative to that literature include (i) its structure as a three station serial queueing network; (ii) its congestion-dependent

service times; and (iii) its susceptibility to congestion-induced collapse. Those features are shared individually with several other papers. For example, the study of admission control into tandem queues goes at least as far back as Ghoneim (1980); Hordijk and Spieksma (1989) also consider an admission control problem for a system with workload-dependent service rates, and in fact study a constrained Markov decision process similar to the one we formulate here; Alderson (2005) studies the control of admissions into simple Markovian queues exhibiting congestion-induced collapse (see also Bekker 2005 and references therein). The work of Duffield and Whitt (1997) on $M/G/\infty$ queues is also complementary to ours, as it focuses on characterizing the dynamics of recovery from high congestion events (in contrast, our control model includes a constraint imposing an upper bound on the probability of experiencing such events). Our queueing model is also related to that developed in Gallien and Wein (2001) to optimize component replenishment in a stochastic assembly system. The analysis in Gallien and Wein (2001) relies on the so-called synchronization assumption that any component may only be assembled with the other components that were ordered from the supplier at the same time it was. While that assumption is difficult to defend in many industrial assembly systems, it holds true for the pick-to-ship process considered here, because in this setting every item is unambiguously assigned to a customer order at the time when it is picked.

Our model, which was developed to provide a realistic representation of a large actual industrial process, is arguably more complex than that presented in all the papers just mentioned. Service times in our queueing network depend on a linear combination of the queue sizes of the first two stations, while the system collapse that we seek to avoid is triggered by the sum of the queue sizes of the last two stations exceeding some threshold – we are not aware of papers studying queueing network models where a function of congestion in some parts of the network similarly affect system dynamics. But we emphasize that our goal and methodology differ substantially from those pursued in all these other papers. Specifically, we seek to compute operational release control policies with a good performance, as opposed to the more ambitious objective of characterizing the structure of the optimal policy for our problem formulation, or even an approximate version of it derived for example through stochastic process limits (see e.g. Harrison 1990). Furthermore, we ultimately resort to numerical and simulation methods, as opposed to the exact mathematical analysis

developed in many of these papers. We still believe that our work may be useful beyond the context of e-tailing warehouses, in particular to those facing other large industrial stochastic control problems that are difficult to address with current analytical methods (see discussion in §4).

2. Model and Analysis

2.1. Model Definition. Our model is a three station serial queueing network defined as follows (one may find it useful to refer to §1.1 when reading the following):

- Each circulating entity in this network represents a customer order, and the release rate of these orders over time is the primary control we seek to optimize. This input is modelled as a Poisson process whose rate at time $\tau \geq 0$ is denoted $\lambda(\tau)$, and the corresponding release control policy will be noted $\boldsymbol{\lambda}$; this is motivated by analytical tractability considerations, and also reflects the uncertainty associated with pickers' productivity as well as other operational factors. Because the maximum release rate achievable is limited in practice by various factors not captured in our model (e.g. number of pickers on staff), we assume an upper bound $\bar{\lambda}$ for this control. Because of our focus on peak demand periods, that upper bound is also assumed exogenous and constant, as the virtual queue of orders placed by customers but not yet released in the warehouse was always significant and never disappeared during such periods. Furthermore, because other practical considerations limit the frequency at which the value of the release rate may be changed our partner felt that implementable policies would only change the release control value 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 release of a customer order into the picking areas 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 as defined in §1.1, 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 orange (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 green (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 $Z_t \triangleq Z(t\delta)$.

Note that the service times A and B of the first and second stations in this model can be expressed in terms of the notation defined in §1.1 as

$$\begin{cases} A = \min\limits_{1\leq i\leq M}(T_i + \ell_i) \\ B = \max\limits_{1\leq i\leq M}(T_i + \ell_i) - \min\limits_{1\leq i\leq M}(T_i + \ell_i) \end{cases}, \tag{3}$$

where M is the number of items per order. However, we emphasize that in the present section these service times are considered given input data (i.e. we do not seek to jointly optimize item release control and order release control). More generally, the key considerations that led us to formulate the specific predictive model just described include:

- The primary model data $(A, B, C, \bar{\lambda}, w)$ is readily available in practice, as described at the end of §1.1;
- The primary model output $(X(\tau), Y(\tau), Z(\tau))$ is directly observable in practice, and in fact that information corresponds exactly to that observed by managers when setting the order release rate. This model feature thus guarantees that the informational requirements

of any release control policy developed will be realistic, and also allows for a quantitative model validation to be performed (see §2.2.2);

- Denoting by n the total number of sorter chutes available, the occurence of gridlock at time τ can be directly expressed in terms of our model's primary output as the event $Y(\tau) + Z(\tau) > n$. Besides, consider any release control policy $\boldsymbol{\lambda}$ ensuring with sufficiently high probability that this event does not occur (we will discuss shortly 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, as expressed by (3). In turn, those transit times are affected in practice by the congestion upstream of the sorter, as discussed in §1.1. 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[3]. 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 the one described above, as it would have to account for merge points, the dynamic tote scheduling logic implemented at the circulating buffer, the item composition of totes, and several other features of the actual pick-to-ship process that are not discussed here in order to protect our industrial partner's confidential information. The method we use instead to address this issue is the same one that we developed for the item release control problem, as described in the Appendix. Specifically, we defined a small number (five) 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 system between the picking area and the sorter. We then proceeded to 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 obtained empirical distributions $A(g)$ and $B(g)$ for all the

[3] 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.

congestion levels g that we had just defined. We were able to construct the congestion levels so that the distributions $A(g)$ and $B(g)$ were unimodal for each g (whereas the empirical distribution of A and B obtained for the entire data set were not), thus validating our approach at a qualitative level (see §2.2.2 for a quantitative validation, and the Appendix for empirical density plots of these service time distributions).

The approach just described also generated interesting new insights about the behavior of the pick-to-ship process. Specifically, while the mean time-to-chute $\mathbb{E}[A(g)]$ followed an unsurprising overall increasing trend with g, the mean chute-dwell time $\mathbb{E}[B(g)]$ exhibited a noticeable drop at an intermediary congestion level, and increased beyond that – see Figure 3 for illustration. Our industrial partner and we believe this phenomenon, which

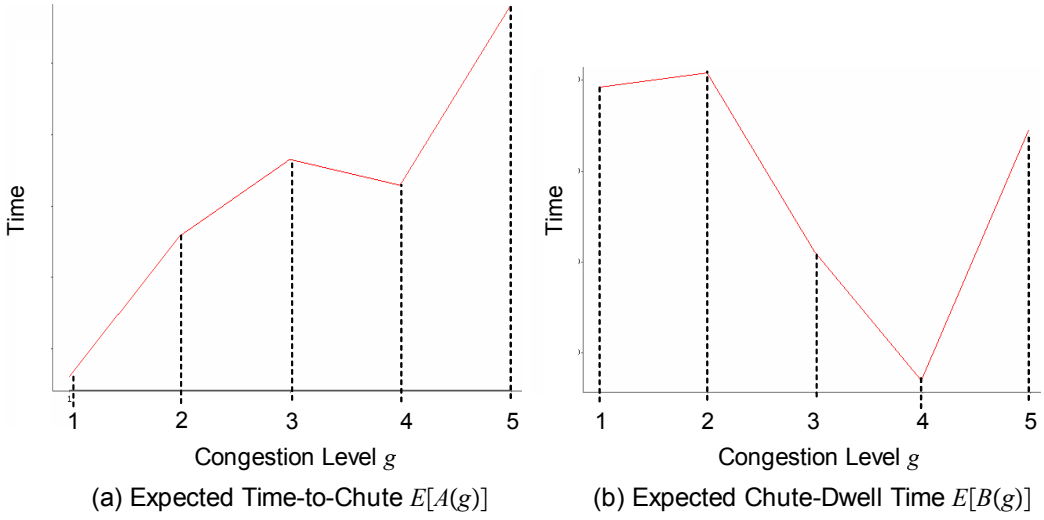


Figure 3: Variation of Expected Time-to-Chute $E[A(g)]$ and Chute-Dwell Time $E[B(g)]$ with Congestion Level g

we consistently observed on several disjoint data sets and with various congestion level definitions, to be explained by the circulating buffer (see §1.1). 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 3 (b)). For medium to high congestion levels (3 and 4 on Figure 3 (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 3 (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 3 (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 3) resulting from the design of this process, further motivating the development of order release control policies able to stabilize the process around it.

Finally, the last modeling step consisted of identifying a relationship between the total number of items $I(\tau)$ on the conveyor system that characterized the congestion level g and the output $(X(\tau), Y(\tau), Z(\tau))$, so that the dynamics of our model would be well defined. With $\mathbb{E}[M]$ denoting the average number of items per customer order as before, the expression $\mathbb{E}[M]X(\tau)$ resulted in a significant underestimation of the number of items $I(\tau)$ 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 $I(\tau) \approx \mathbb{E}[M](X(\tau) + Y(\tau)/2)$ provided 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 was validated by performing a comparison of actual system output and simulated model output over time for given starting conditions and release rate history (see §2.2.2).

2.2. Approximate Dynamics.

2.2.1. Derivation. Our next step is to study the dynamics of the queueing model described in §2.1, 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 in §1.1. 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 §2.3), 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 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 beyond three of the number of state space dimensions does increase computational requirements for our approximate DP algorithm to a level that is unpractical, 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 expected service times $\mathbb{E}[A(g)]$ and $\mathbb{E}[B(g)]$ at the first and second stations obtained from data are several times larger than the control period δ, the transitions that this assumption allows to ignore 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 §2.1 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):

$$\left\{\begin{array}{l} X_{t+1} = X_t + N_t^{\rightarrow X} - N_t^{X\rightarrow Y} \\ Y_{t+1} = Y_t + N_t^{X\rightarrow Y} - N_t^{Y\rightarrow Z} \\ Z_{t+1} = Z_t + N_t^{Y\rightarrow Z} - N_t^{Z\rightarrow} \\ g_t = \sum_{g=1}^{\bar{g}} g 1_{[d_g, d_{g+1})}(\mathbb{E}[M](X_t + \frac{Y_t}{2})) \end{array}\right. \quad \text{with} \quad \left\{\begin{array}{l} N_t^{\rightarrow X} \sim \text{Poisson}(\lambda_t \delta) \\ N_t^{X\rightarrow Y} \sim \text{Binom}(X_t, 1 - e^{-\frac{\delta}{\mathbb{E}[A(g_t)]}}) \\ N_t^{Y\rightarrow Z} \sim \text{Binom}(Y_t, 1 - e^{-\frac{\delta}{\mathbb{E}[B(g_t)]}}) \\ N_t^{Z\rightarrow} \sim \text{Poisson}(\frac{(w \wedge Z_t)\delta}{\mathbb{E}[C]}) \end{array}\right. , \tag{4}$$

where the four random variables $(N_t^{\rightarrow X}, N_t^{X\rightarrow Y}, N_t^{Y\rightarrow Z}, N_t^{Z\rightarrow})$ represent the number of customer orders that are respectively released into the first station, moved from the first to the second and the second to the third station, and processed out of the third station, between time periods t and $t+1$. An appealing feature is that simulating system (4) 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 $N_t^{X\rightarrow Y}$ and $N_t^{Y\rightarrow Z}$ 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.

2.2.2. Validation. Our next step was to validate the approximate queueing dynamics (4) 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, 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 simulated the random variables $(X_{t+1}, Y_{t+1}, Z_{t+1})$ characterized by (4) 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 4 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 4 (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 4 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 (see §1.1). 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 induc-

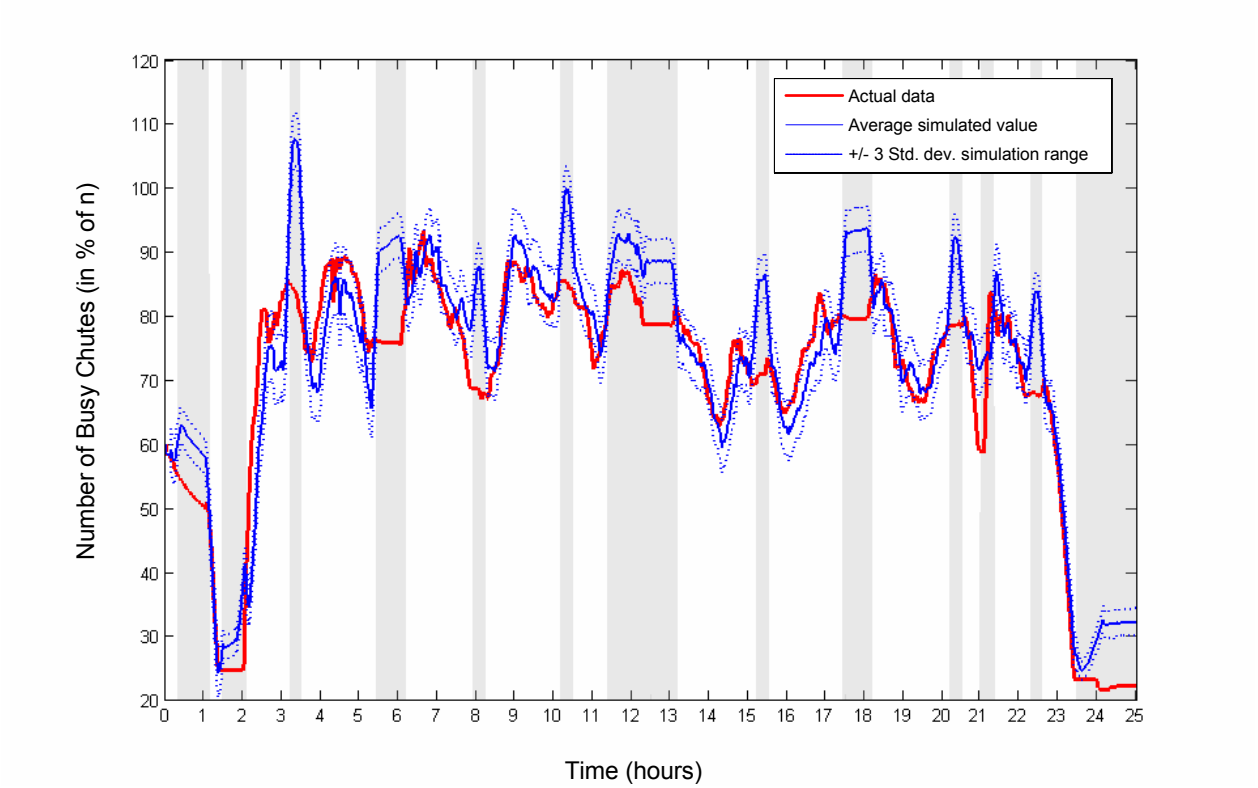


Figure 4: Predicted Distribution $Y_{t+1} + Z_{t+1}$ and Actual Value $y^*(\tau + \delta) + z^*(\tau + \delta)$ of the Number of Busy Chutes over a 24 Hour Period

tion stations. This modeling choice is justified during the regular (non-highlighted) working hours, as the induction stations have ample spare processing capacity then, and thus do not have a significant impact on the transit times of items between the picking areas and the sorter. However, the periods highlighted in Figure 4 do impact the staffing of these induction stations, so that the actual flow of items into the sorter then is either drastically reduced (maintenance/breakdown) or stopped (work breaks). Indeed, note that the actual number of busy chutes remains constant during all work breaks listed above, which reflects that 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 our partner was not concerned by throughput performance or the risk of

gridlock during periods of forced reduced activity. Because in contrast the actual number of busy chutes observed during regular working hours almost always lied within our model's predicted range, the approximate dynamics tested appeared sufficiently accurate given our purposes, and this validation exercise was deemed conclusive.

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

$$\begin{array}{lll} CDP[\beta]: & \max\limits_{\boldsymbol{\lambda}} & \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t \lambda_t | X_0, Y_0, Z_0] \\ & \text{s.t.:} & \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\left\{Y_t^{\boldsymbol{\lambda}} + Z_t^{\boldsymbol{\lambda}} > n\right\}} | X_0, Y_0, Z_0] \leq \beta \\ & & \lambda_t \in [0, \bar{\lambda}] \text{ for all } t \in \mathbb{N}, \end{array} \tag{5}$$

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 (5), 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 (5) 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 §2.1. 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 (5) 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 (5) through the following limiting statements[4], which are proven in Blackwell (1962) and hold for any initial state:

$$\begin{cases} \lim_{\alpha\to 1-}(1-\alpha)\mathbb{E}[\sum_{t=0}^{+\infty}\alpha^t\lambda_t|X_0,Y_0,Z_0] = \lim_{k\to\infty}\frac{1}{k}\mathbb{E}[\sum_{t=0}^{k-1}\lambda_t] \\ \lim_{\alpha\to 1-}(1-\alpha)\mathbb{E}[\sum_{t=0}^{+\infty}\alpha^t 1_{\{Y_t+Z_t>n\}}|X_0,Y_0,Z_0] = \lim_{k\to\infty}\frac{1}{k}\mathbb{E}[\sum_{t=0}^{k-1}1_{\left\{Y_t^{\boldsymbol{\lambda}}+Z_t^{\boldsymbol{\lambda}}>n\right\}}] = \lim_{t\to\infty}\mathbb{P}(Y_t+Z_t>n) \end{cases}. \tag{6}$$

The second and third equality statements in (6) justify an interpretation of the first constraint in (5) as an upper bound on the steady-state probability that the system is in a state of gridlock (i.e. the last term in (6)). Specifically, that upper bound is close to $(1-\alpha)\beta$ when the discounting factor α is close to 1, where β is the risk budget appearing in (5). From a modeling perspective, that constraint thus indeed balances the throughput maximization objective in (5) with the need to avoid the state of gridlock. We emphasize however that the fairly unintuitive expression of that constraint in (5) (i.e. a discounted sum of indicator functions) is only motivated by technical dynamic programming considerations (see §2.4). Also, that constraint 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), and so any policy we could derive by solving that formulation could conceptually depend on it as well. 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 (6), where the r.h.s is independent of the initial state, as is typical of the objective of an average cost DP formulation.

2.4. Numerical Optimization Methods. We have specifically formulated the dynamic program $CDP[\beta]$ in (5) so it would belong to a family of constrained Markov decision processes for which theoretical results and computational methods can be easily derived (see Altman 1999 for a review). In particular, we now outline a method for computing a solution to $CDP[\beta]$ by solving a sequence of related unconstrained dynamic programs

[4] 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.

$UDP[\theta]$ obtained for any $\theta \geq 0$ as

$$UDP[\theta]: \quad \max_{\boldsymbol{\lambda}} \quad \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t \left(\lambda_t - \theta.1_{\{Y_t+Z_t>n\}}\right) | (X_0, Y_0, Z_0) = (x, y, z)] \\ \text{s.t.:} \quad \lambda_t \in [0, \bar{\lambda}] \text{ for all } t \in \mathbb{N}, \tag{7}$$

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 (5) 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 $\mathbf{c}^\theta(x, y, z) \triangleq \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\left\{Y_t^{\boldsymbol{\lambda}^\theta} + Z_t^{\boldsymbol{\lambda}^\theta} > n\right\}} | (X_0, Y_0, Z_0) = (x, y, z)]$, where $\boldsymbol{\lambda}^\theta$ is an optimal policy for $UDP[\theta]$. The following results are obtained through 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 $\mathbf{c}^\theta$, 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 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, given our relatively modest goal of identifying through numerical methods good policies for the order release control problem facing our industrial partner (as opposed to say computing provably optimal or near-optimal solutions), we were able to reduce computations to a practical level through the following additional steps:

- We use the approximate queueing dynamics described in §2.2.1 when solving each instance of $UDP[\theta]$;

- 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);

- 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 $UDP[\theta^k]$.

The next section contains a discussion of the qualitative features and quantitative performance of the policies computed with the numerical methods just described.

3. Numerical Experiments

The goal of our simulation study is to estimate in various environments and understand the relative performance of the order release control policy derived through the model and numerical optimization methods described in §2 (hereafter denoted by ADP and its release rate function $\lambda^{ADP}(x,y,z)$) against that of other possible policies. In addition to the release control policy used by our industrial partner in the past, we specifically consider:

Policy CST (constant release): Releases orders at the constant rate $\lambda^{CST} \in [0, \bar{\lambda}]$ corresponding to the best constant solution to (5). That rate is easily found by simulation-based search.

Policy CWP (constant work in process): Releases orders at a rate given by the func-

tion

$$\lambda^{CWP}(x,y,z) = \begin{cases} \bar{\lambda}^{CWP} \text{ if } x+y+z < \bar{k} \\ 0 \text{ otherwise} \end{cases}, \tag{8}$$

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 (5) within the family of CONWIP policies defined by (8). We consider CONWIP policies here because despite their simplicity, they have been found to perform well in many environments (Spearman and Zazanis 1992).

We first consider in §3.1 the performance of these policies in steady-state, then in §3.2 their transient performance when subjected to various transient disruptions.

3.1. Steady-State Simulation Experiments. The range of simulation scenarii we consider is characterized by five different values for the number of packers $w \in \{75\%p,$ $87.5\%p$, p, $112.5\%p$, $125\%p\}$, where p denotes the average number of packers working in our industrial partner's warehouse during a peak demand period, under current staffing policies. 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 §2.3). 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 $D \in \{ADP, CWP, CST\}$, where the notation λ_t^D abbreviates $\lambda^D(X_t, Y_t, Z_t)$ 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). Note that for policy CST this quantity is simply λ^{CST}, and that an upper bound for the throughput rate of all policies is the packing capacity $w(\mathrm{E}[C])^{-1}$ (or service rate of the third station described in §2.1). In order to enable a meaningful assessment of the improvement potential associated with the implementation of these policies in our partner's warehouse (and preserve its confidential information), all throughput and rate data is provided as a ratio to the average throughput γ observed in our industrial partner's warehouse during a period with no breaks and p packers assigned to the sorter. Table 1 summarizes our steady-state simulation results[5].

[5] Table 1 notes: All results shown correspond to the low risk level $\underline{\beta}$, except for the numbers in brackets which were obtained with $\bar{\beta}$. All numbers shown in the third and subsequent rows are percentages. All simulation results reported have a standard estimation error smaller than 0.2%.

Number of Packers w	$75\%p$	$87.5\%p$	p	$112.5\%p$	$125\%p$
Chutes per Packer n/w	73.3	62.8	55	48.8	44
Relative Packing Capacity $w(\mathbb{E}[C])^{-1}/\gamma$	78.8	91.9	105	118	131.2
γ^{ADP}/γ	78.7 [78.7]	91.5 [91.5]	104.4 [104.4]	109.3 [109.5]	110.3 [110.4]
$\gamma^{ADP}/w(\mathbb{E}[C])^{-1}$	99.9 [99.9]	99.6 [99.6]	99.4 [99.4]	92.6 [92.8]	84.1 [84.1]
$\frac{\mathbb{E}[X_\infty]}{n}/\frac{\mathbb{E}[Y_\infty]}{n}/\frac{\mathbb{E}[Z_\infty]}{n}$	62.3/55.3/17.9	72.7/59.7/19.9	82.9/66.3/19.2	87/69.7/8.5	87.7/70.1/8.9
γ^{CWP}/γ	78.3 [78.3]	91.3 [91.3]	104 [104.3]	105.3 [106.4]	105.8 [106.5]
$\gamma^{CWP}/w(\mathbb{E}[C])^{-1}$	99.3 [99.3]	99.4 [99.4]	99.0 [99.3]	89.2 [90.1]	80.6 [81.2]
$\frac{\mathbb{E}[X_\infty]}{n}/\frac{\mathbb{E}[Y_\infty]}{n}/\frac{\mathbb{E}[Z_\infty]}{n}$	62.9/55.4/33.3	72.9/59.5/28.7	83.1/66.5/13.9	85.5/68.7/8.4	85.6/68.8/8.3
λ^{CST}/γ	77.1 [77.7]	89.5 [90.7]	101.9 [103]	102.7 [104.6]	102.8 [104.7]
$\gamma^{CST}/w(\mathbb{E}[C])^{-1}$	97.9 [98.6]	97.4 [98.8]	97.0 [98.0]	87.0 [88.7]	78.3 [79.8]
$\frac{\mathbb{E}[X_\infty]}{n}/\frac{\mathbb{E}[Y_\infty]}{n}/\frac{\mathbb{E}[Z_\infty]}{n}$	62.8/55.5/8.1	70.9/56.9/8.3	80.7/64.5/9.8	81.3/65.9/8.2	83.0/66.4/8.5

Table 1: Steady-State Simulation Results for Order Release Control Experiments

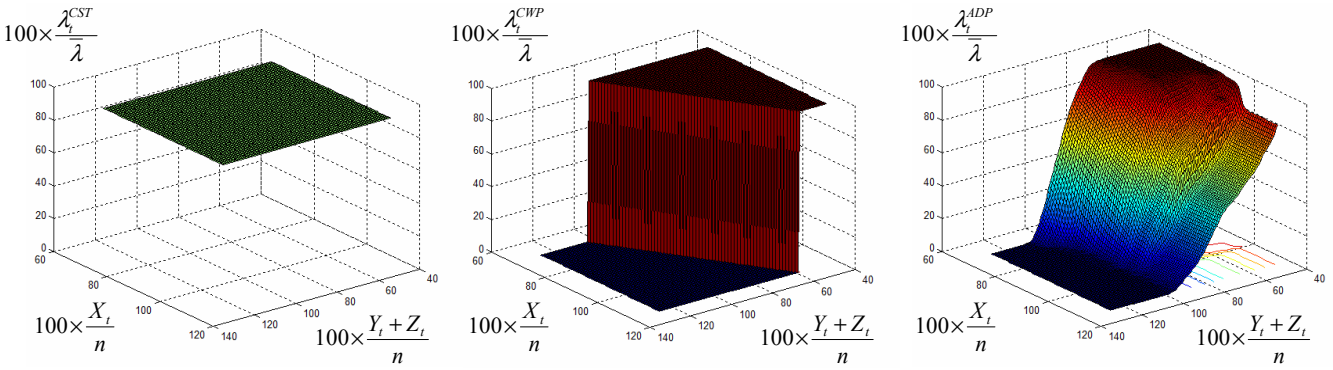
Table 1 shows that for every policy $D \in \{ADP, CWP, CST\}$ considered, the effective packing capacity utilization $\gamma^D/w(\mathbb{E}[C])^{-1}$ remains almost constant at 97% and above as the number of packers w increases from $75\%p$ to p, then suddenly drops to 92.6% and below for $w = 112.5\%p$ and even more drastically at 84.1% and below for $w = 125\%p$. 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 scenarii 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 ($w \geq p$) 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 then

the apparent 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 and CWP in the high congestion scenarii where $w \leq p$.

We 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. 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 system-wide impact of local staffing policies.



(a) Policy CST (b) Policy CWP (c) Policy ADP

Figure 5: Release Rate Function of Policies ADP, CWP and CST for the Scenario $(w,\beta)=(p,\underline{\beta})$

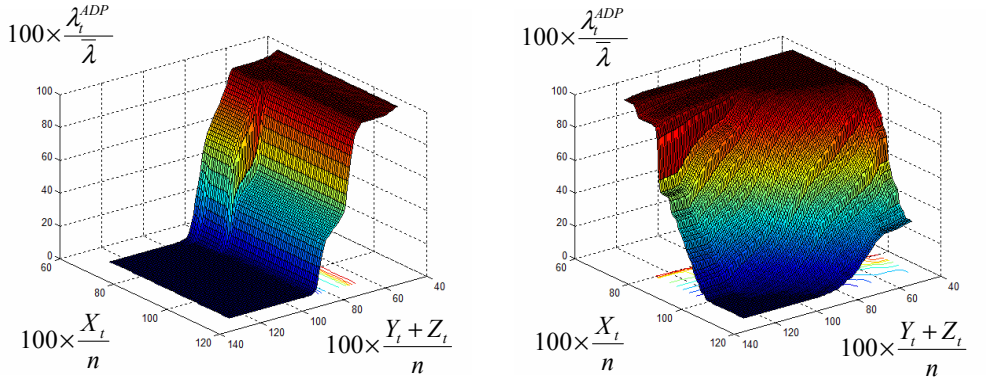
Table 1 also enables a performance comparison across policies. It suggests that the long-run throughput of ADP dominates that of CWP, which itselfs dominates that of CST. However, those relative superiorities increase with the number of packers w: The performances of all three policies are all relatively close for the lowest number of packers considered ($75\%p$), the throughput superiority of ADP relative to CWP only becomes larger than 1% in our experiments for $w>p$ (it reaches 4% for $w=112.5\%p$ and 4.5% for $w=125\%p$), and that of CWP relative to CST increases from 1.2% to 3% as w increases from $75\%p$ to $125\%p$. Figure 5, which contains graphs representing the functions $\lambda^D(X_t,Y_t,Z_t)$ in the plane (X_t,Y_t+Z_t) for each policy $D\in\{ADP,CWP,CST\}$[6] in the scenario $(w,\beta)=(p,\underline{\beta})$, illustrates that the performance differentials observed across the three policies considered can be explained by their different abilities to finely adapt the instantaneous release rate to the system state observed at each control epoch. In particular, ADP and to a lesser extent CWP are able to address any temporary stochastic increase of the number of orders in transit X_t or busy chutes Y_t+Z_t above their implicit operating averages by reducing the instantaneous release

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

rate accordingly, which allows them to mitigate such increase faster and ultimately maintain a higher average throughput for the same level of risk. Conversely, ADP and to a lesser extent CWP better exploit temporary stochastic decreases of X_t or $Y_t + Z_t$ below the ideal steady-state by instantaneously releasing more orders then, which also contributes to their higher long-term throughput. The fact that congestion is more sensitive to variations in the release rate in the scenarii with a low number of packers that are heavily utilized (see earlier remarks on the results for $\beta = \bar{\beta}$) also explains that the performance differentials between the policies considered increase with w. Specifically, the range of instantaneous release rates that do not lead to a violation of the gridlock probability constraint is more limited when packing utilization is high. As a result, the ability of CWP and ADP to dynamically adapt the release rate to process conditions does not provide substantial benefits then (Table 1 shows that CST is less than 3% suboptimal in all scenarii with $w \leq p$), in contrast with situations where packing is less utilized ($w > p$).

Because of its considerably richer structure however, ADP is able to respond more finely to changes in individual features of the system state than CWP, which is only sensitive to the total number of orders in the system $X_t + Y_t + Z_t$. A first important reason why this difference is material is that the system dynamics (specifically the service times of the first and second stations in our model) depend on the congestion level of the conveyor system, itself proportional to the state quantity $X_t + Y_t/2$ (see §2.1). Policy ADP is thus able to specifically control the system so as to maintain it in a desirable congestion level (e.g. $g = 4$ in Figure 3), whereas CWP does not differentiate between two system states corresponding to possibly very different congestion levels $X_t + Y_t/2$ as long as they share the same value of $X_t + Y_t + Z_t$. This feature of ADP and the fact that transitions between different congestion levels g are discrete in our model (see §2.1) explain the somewhat abrupt change of the release rate λ^{ADP} around $X_t/n \approx 110\%$ seen in Figure 5 (c). A second reason why this richer structure is material is that it enables ADP to adapt its behavior to different staffing levels for the number of packers. This is illustrated by Figure 6, which represents ADP's release rate functions for the scenarii $(w, \beta) = (75\%p, \underline{\beta})$ and $(125\%p, \underline{\beta})$ corresponding to the lowest and highest numbers of packers respectively, in the same format as used in Figure 5.

Figure 6 (a) shows that with a small number of packers, the instantaneous release rate



(a) w = 75% p (b) w = 125% p

Figure 6: Release Rate Function of Policy ADP for the Scenarii $(w, \beta) = (75\%p, \underline{\beta})$ and $(125\%p, \underline{\beta})$

$\lambda^{ADP}(X_t, Y_t, Z_t)$ is almost independent of X_t (with the exception of the abrupt change around $X_t/n \approx 90\%$, which corresponds to a transition between two consecutive congestion levels). As discussed above, in this regime any policy with a high throughput must heavily utilize packing capacity, which is captured by the third station in our model. Consequently, any given change in the arrival rate of orders to that station has a substantial impact on its occupancy process, or number of green chutes Z_t. The ADP policy (which in this scenario is less than 0.05% suboptimal) thus reacts considerably more to changes in the state variable Z_t than to changes of X_t or Y_t. In addition, it compensates 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 6 (a) around $(Y_t + Z_t)/n \approx 80\%$. In contrast, Figure 6 (b) illustrates that with a high number of packers, policy ADP reacts much more to changes in the number of orders in transit X_t than to changes in the number of busy chutes $Y_t + Z_t$; other representations (not included here) show that $\lambda^{ADP}(X_t, Y_t, Z_t)$ is in fact almost independent of Z_t then. In this scenario with low packing utilization, green 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 §2.1).

Observe that the ADP policy shown in Figure 5 (c) thus exhibits an intermediate behavior between the two extremes seen in Figures 6 (a) and (b). Together, these Figures illustrate how policy ADP adapts to different staffing levels, and the results shown in Table 1 suggest that the effect of this feature on throughput performance may be significant.

Finally, the results shown in Table 1 suggest that, in the case where p packers are assigned to the sorter, policies CST, CWP and ADP 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 conditions. It may be surprising at first that even policy CST outperforms the policy used in our partner's warehouse. We point out however that despite its simplicity CST 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 and our partner's historical policy would be maintained in such scenarii, we can speculate from Table 1 that policy ADP (resp. CWP) 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 and addition of 25% more packers than p would increase throughput relative to the historical performance we have observed by about 10.3%. Table 1 also suggests that the rationale underlying the informal guidelines used historically to control order release in our partner's warehouse, namely empirical targets for the number of orange and green chutes, may not be a sound one. Specifically, our simulation estimates for each policy and scenario of the average system state in steady state $(\mathbb{E}[X_\infty], \mathbb{E}[Y_\infty], \mathbb{E}[Z_\infty])$ (shown in Table 1 as fractions of the number of sorter chutes n) indicate that the scenarii for which the average operating states differ most across policies ($w \leq 87.5\%p$) coincide with those for which the policies' performance differentials are smallest. Conversely, the differences in throughput across policies are highest in the scenario $w = 125\%p$, where their steady-state average operating states seem most similar. It thus seems that the performance differentials across policies are less driven by their average operating states than by how each policy responds

	No Disruption	Experiment 1	2	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 2: Gridlock Probabilities During Transient Simulation Experiments

to variations of the system state from that average point.

3.2. Transient Simulation Experiments. The goal of our second set of simulation experiments was to assess the robustness of the policies considered relative to temporary mispecifications of the input data under which they are derived. 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 simulation 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 2 contains a summary of our results[7], which we discuss in the remainder of this section after a more detailed description of each experiment.

[7] Table 2 notes: All results are shown as percentages, and have a standard estimation error from simulation lower than 0.5%.

3.2.1. 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 to gridlock.

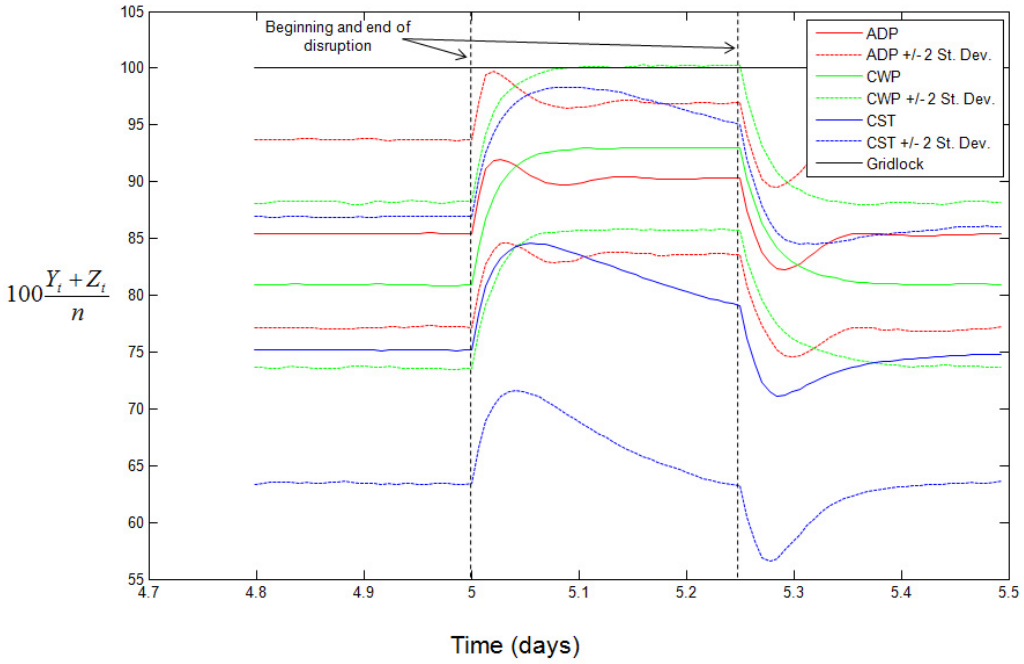


Figure 7: Evolution of the Average Fraction of Busy Chutes $\frac{Y_t+Z_t}{n}$ during Experiment 1

Figure 7 represents the evolution over time of the fraction of busy chutes (averaged across replications) for all three policies considered, while Figure 8 represents an average of their release rates over the same time period. Observe first (from the period prior to the disruption shown in Figure 7) 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

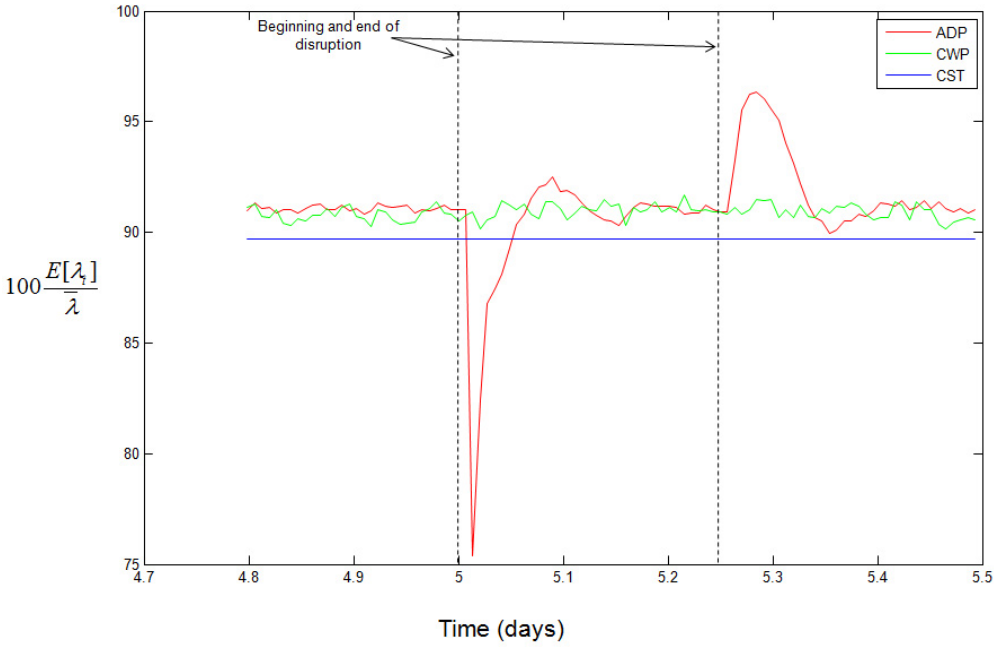


Figure 8: Evolution of the Average Relative Release Rate $\frac{E[\lambda_t]}{\bar{\lambda}}$ during Experiment 1

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 to gridlock, which it only experienced in 2.6% of all simulated replications (see Table 2). As the disruption ends, the second queue is suddenly starved, which ADP sees as an opportunity to release more orders (as seen in Figure 8), 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 8. 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 7). 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 2).

3.2.2. 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 9 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

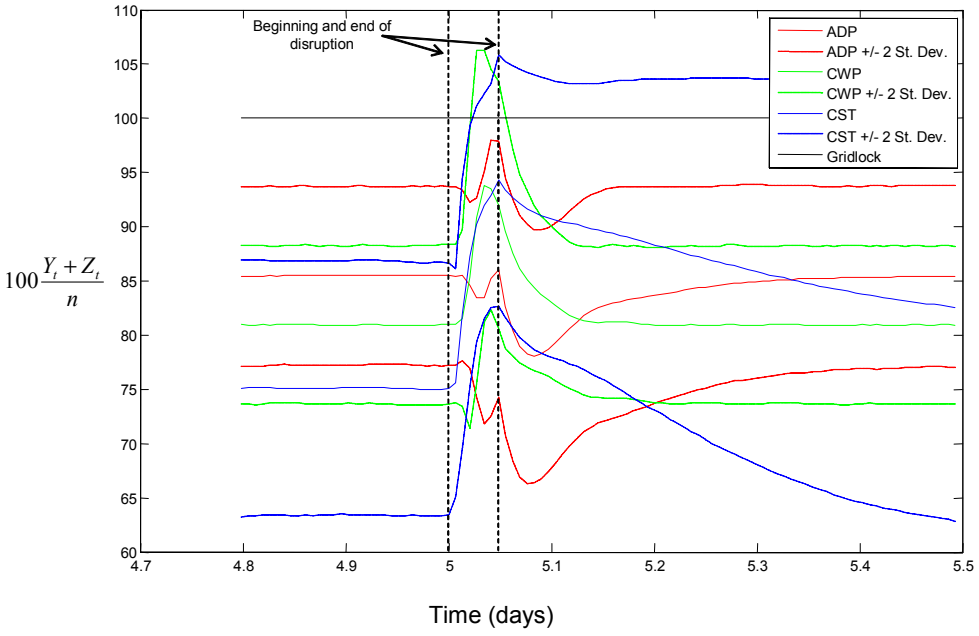


Figure 9: Evolution of the Average Fraction of Busy Chutes $\frac{Y_t+Z_t}{n}$ during Experiment 2

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 9). Finally, the convergence of the system back to 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 9), and CST performs worst overall in that experiment among all policies considered (as seen in Table 2 CST^{β} experienced gridlock in 47.1% of replications, CST^{γ} in 94%).

As seen in Figure 10 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 9). 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

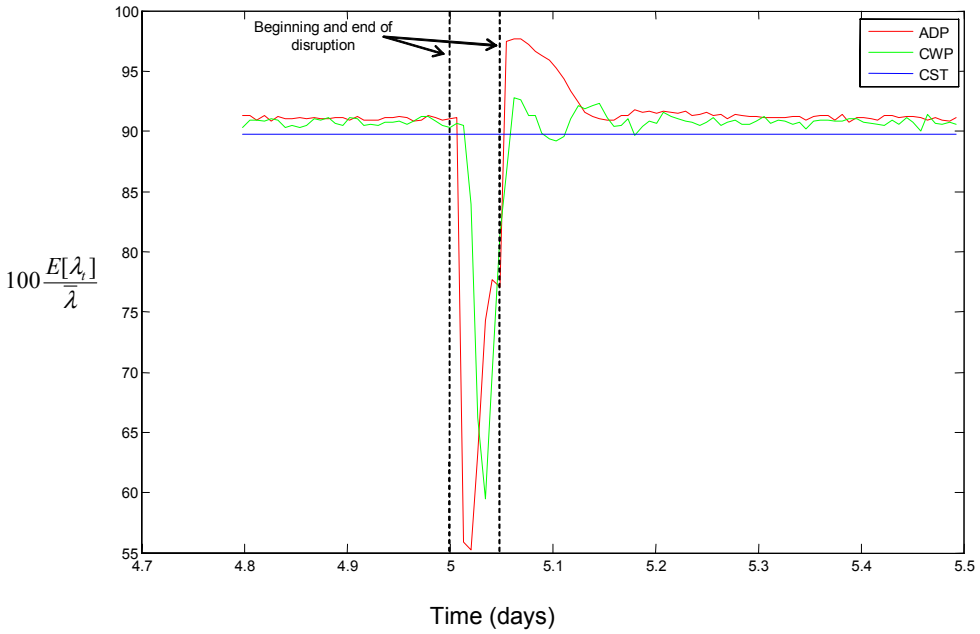


Figure 10: Evolution of the Average Relative Release Rate $\frac{E[\lambda_t]}{\bar{\lambda}}$ during Experiment 2

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 seen in Table 1 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 $X_t + Y_t + Z_t$, 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 2 CWP^β entered gridlock in 39% of replications, CWP^γ in 68%).

As seen in Figure 10, 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 (from Table 1, $\frac{\mathbb{E}[Z_\infty]}{n} = 19.2\%$ 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 9 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 2). 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 9 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 10 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 2).

3.2.3. 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 §3.1), that disruption turns out to be quite severe.

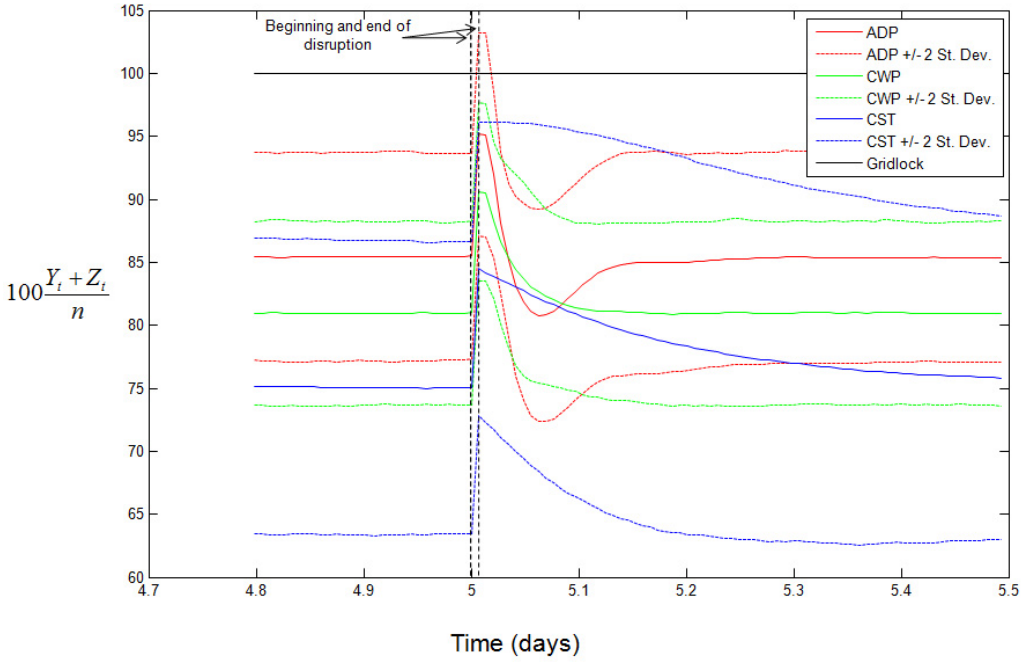


Figure 11: Evolution of the Average Fraction of Busy Chutes $\frac{Y_t+Z_t}{n}$ during Experiment 3

Indeed, Figure 11 shows that under all policies considered the average number of busy 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 2 (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%

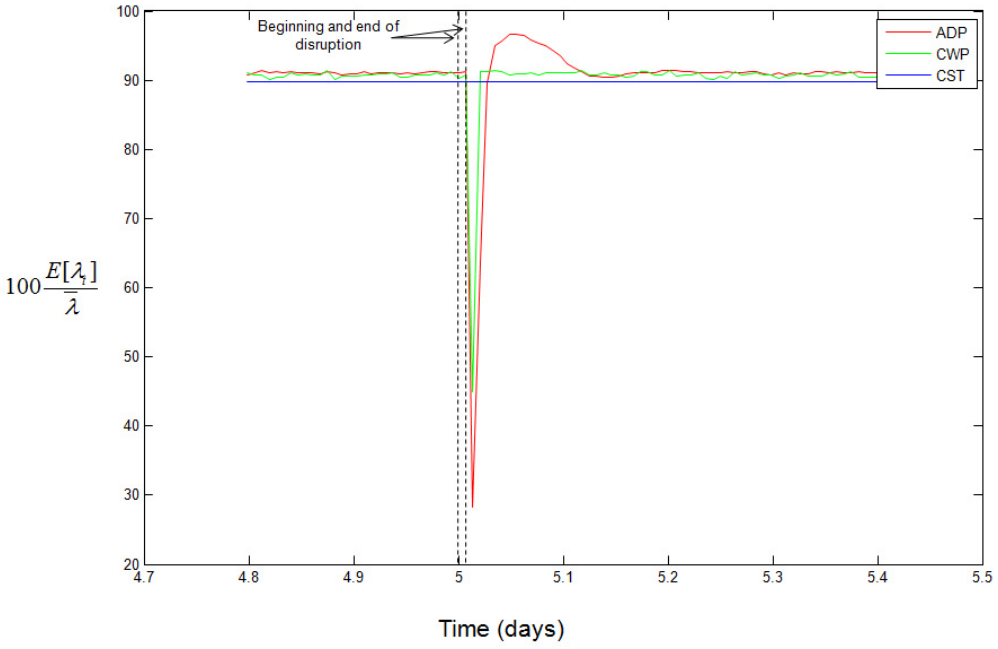


Figure 12: Evolution of the Average Relative Release Rate $\frac{E[\lambda_t]}{\bar{\lambda}}$ during Experiment 3

respectively, as seen in Table 1) 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 11 to increase over the disruption period by approximately 9.5% of the total number n available, regardless of the policy considered. 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 2 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}).

4. Conclusion

Our results suggest that the performance of the specific pick-to-ship process under study, arguably the industrial process that is most critical to the fulfillment of online consumer orders on a large scale, may be substantially improved through better order release control policies of the type we have derived here using numerical methods. Specifically, simulation experiments discussed in §3 indicate that an implementation of our policy (ADP) would likely yield a higher throughput than the policy used by our industrial partner at the outset of

our collaboration as well as other simple possible policies for this problem such as CONWIP (CWP) and constant release (CST). To the extent that the performance superiority of ADP relative to all other policy considered does vary (increase) with the number of packers (see §3.1), these experiments also shed light on the impact of local staffing practices on process-wide performance. Perhaps the most striking quantitative prediction of this study from a practical standpoint is that an increase of staffed packers by 25% along with the implementation of our proposed release control policy could increase process throughput by as much as 10% (see discussion in §3.1 for appropriate qualifications of this statement).

Our transient simulation experiments also suggest that ADP is substantially more robust than both CWP and CST with respect to temporary mispecifications of the time-to-chute and chute-dwell time (which corresponds to the service times of the first and second stations in our model, respectively). For these, the more sophisticated structure of ADP derived for a well-specified model does endow that policy with more robustness to a mispecified one. When it is the number of packers which is mispecified however, the transient risk of gridlock associated with ADP is higher than that of the policies CWP and CST with a comparable nominal steady-state risk (that is assuming a well-specificed model, see Table 2). This is because the better flexibility of ADP allows it to operate closer to gridlock for the same level of nominal risk, and mispecifications of the number of packers are hardest to correct because of the control lag associated with the cycle time of orders through the entire process. Note however that the transient probability of gridlock for ADP in those conditions is substantially lower than that of the policies CWP and CST with the same steady-state throughput, so that the performance of ADP along this dimension does not seem particularly alarming. However, these results still suggest that appropriate monitoring of packers' activity is quite important in this setting.

While the present study did have an impact on the operations of our industrial partner, we are unfortunately not at liberty to provide a more detailed account here. In any case, future collaborative work could include the development of order release policies that would discriminate between several classes of orders corresponding to different customer service levels and/or different numbers of items, and may thus more finely capture the benefits and costs associated with processing specific orders. More broadly, while some of our model features may be specific to the e-tailing industry, these models should also apply at least

in part to other industrial warehouses using an automated sorter. In addition, while this study concerns a fairly specific setting, some of the insights it reveals about performance drivers of release control policies may be more widely applicable. The first one is that the improvement potential associated with sophisticated dynamic release policies may be larger in systems where capacity utilization is relatively low. The second is that the throughput performance of release policies may sometimes be less driven by the state in which they keep the system than by how they respond to transient variations around that state (the fact that some managers working with our partner were using a control policy based on target values for the system state at the outset of our interaction suggests that this may not be entirely intuitive). Finally, even though CONWIP is a simple policy that seems effective in a wide variety of systems (see e.g. Spearman and Zazanis 1992), it might be significantly improved upon by more sophisticated policies when congestion in some parts of the system is more significant than in other parts (as is the case here because of the gridlock phenomenon and endogeneous service times).

From a methodological perspective, we also believe that our work may be useful to Operations Research practitioners beyond the context of warehouses, as an example of how operational solutions to a large and complex stochastic industrial control problem, seemingly beyond the reach of current analytical methods, may still be computed numerically using queueing theory and approximate dynamic programming. We also hope that it will provide theoreticians with a well-grounded motivation for developing methods allowing to derive analytical solutions to such problems in the future.

Acknowledgment. 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, 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 Katherine Miller for providing feedback on an earlier version of

this paper. The MIT Leaders For Manufacturing Program has provided invaluable logistical 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

Alderson, D. (2005). Avoiding collapse in congestion-sensitive processing systems. Technical report, California Institute of Technology.

Allgor, R., S. Graves, and P. J. Xu (2004). Traditional inventory models in an e-retailing setting: A two-stage serial system with space constraints. In *Proceedings of the 2004 SMA Conference*, Singapore.

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

Bartholdi, J. J. and S. T. Hackman (2006). *Warehouse and Distribution Science*. in preparation.

Bekker, R. (2005). Finite-buffer queues with workload-dependent service and arrival rates. *Queueing Systems 50*, 231–253.

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.

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.

Duffield, N. G. and W. Whitt (1997). Control and recovery from rare congestion events in a large multi-server system. *Queueing Systems 26*, 69–104.

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

Faranca, A. G. (2004). Complex system analysis through discrete event simulation. Master's thesis, Massachusetts Institute of Technology.

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.

Ghoneim, H. (1980). *Optimal Control of Arrivals to a Network of Two Queues in Series*. Ph. D. thesis, North Carolina State University.

Harrison, J. M. (1990). *Brownian Motion and Stochastic Flow Systems*. Malabar, Florida: Robert E. Krieger.

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

Jackson, D. O. (2005). Managing and scheduling inbound material receiving at a distribution center. Master's thesis, Massachusetts Institute of Technology.

Lieu, C. A. (2005). Impact of inventory storage schemes on productivity. Master's thesis, Massachusetts Institute of Technology.

Price, C. S. (2004). Systems analysis and optimization through discrete event simulation at amazon.com. Master's thesis, Massachusetts Institute of Technology.

Rubenstein, L. (2006). Intelligent product placement strategies for amazon.com's worldwide fulfillment centers. Master's thesis, Massachusetts Institute of Technology.

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

Stidham, S. (1985). Optimal control of admission to a queueing system. *IEEE Transactions on Automatic Control AC-30*(8), 705–713.

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

Xu, P. J. (2005). *Order Fulfillment in Online Retailing: What Goes Where*. Ph. D. thesis, Massachusetts Institute of Technology.

Zeppieri, M. A. (2005). Predictive capacity planning modeling with tactical and strategic applications. Master's thesis, Massachusetts Institute of Technology.