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

Jérémie Gallien[1] and Théophane Weber[2]

May 14, 2007

A.1. Warehouse Outbound Process Description

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

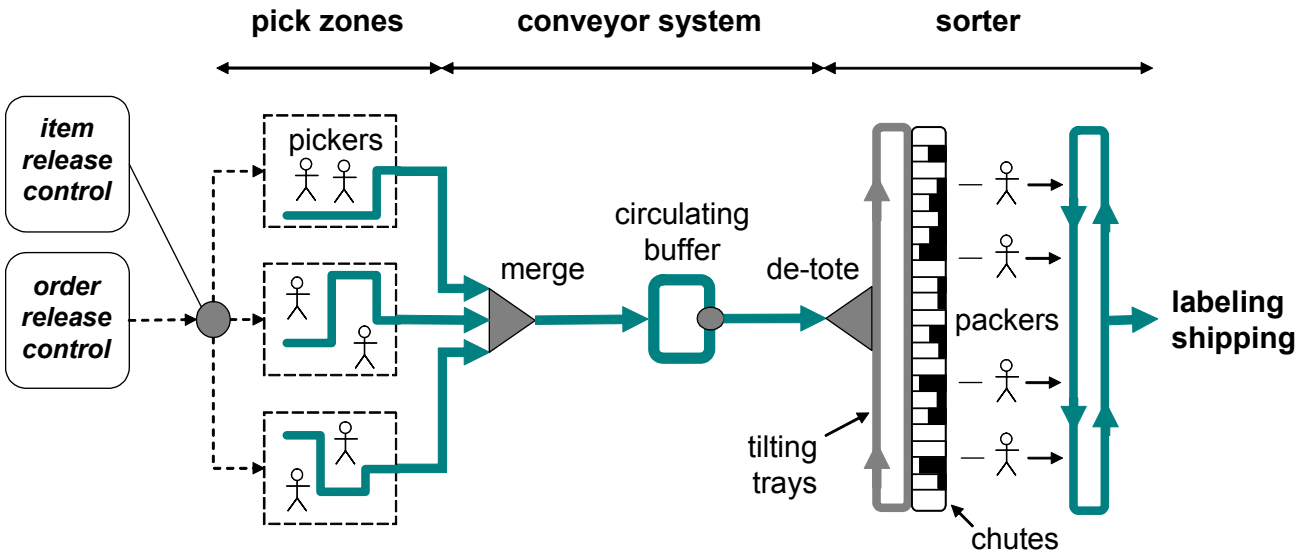


Figure A.1: Flow Diagram of the Pick-to-Ship Process Considered

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

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

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

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

A.2. Flow Data Analysis.

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

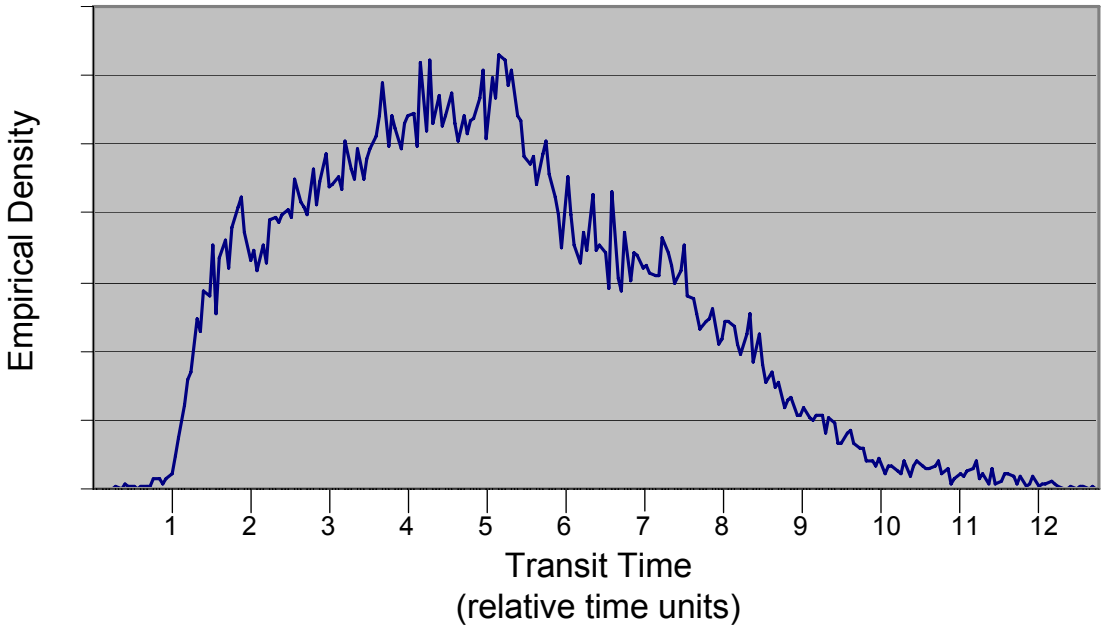


Figure A.2: Empirical Density of Transit Times from the Same Picking Zone Over 24 Hours

Note that the relative time units used for the x-axis in Figure A.2 to disguise our partner's confidential data show a variation from 1 to 12 of the item transit times over that period, which was typical across all picking zones over that time period. Another typical feature is that the distribution shown in Figure A.2 is multi-modal, suggesting that it results from the superposition of several heterogeneous system behavior modes. We hypothesized that

these behavioral modes were primarily driven by conveyor congestion, and that the overall behavior or the pick-to-ship process could be characterized fairly accurately using a limited number of congestion levels, each corresponding to a range of values for the total number of items on the conveyor system between the picking area and the sorter. To verify that hypothesis, we constructed and plotted the data series representing the number of items on the entire conveyor system over time during the same 24h period, and we defined a limited number of congestion levels based on the amount of data available (indexed in the following as $g \in \{1, ..., \bar{g}\}$). Figure A.3 illustrates this process on a dataset which led us to define 7 congestion levels.

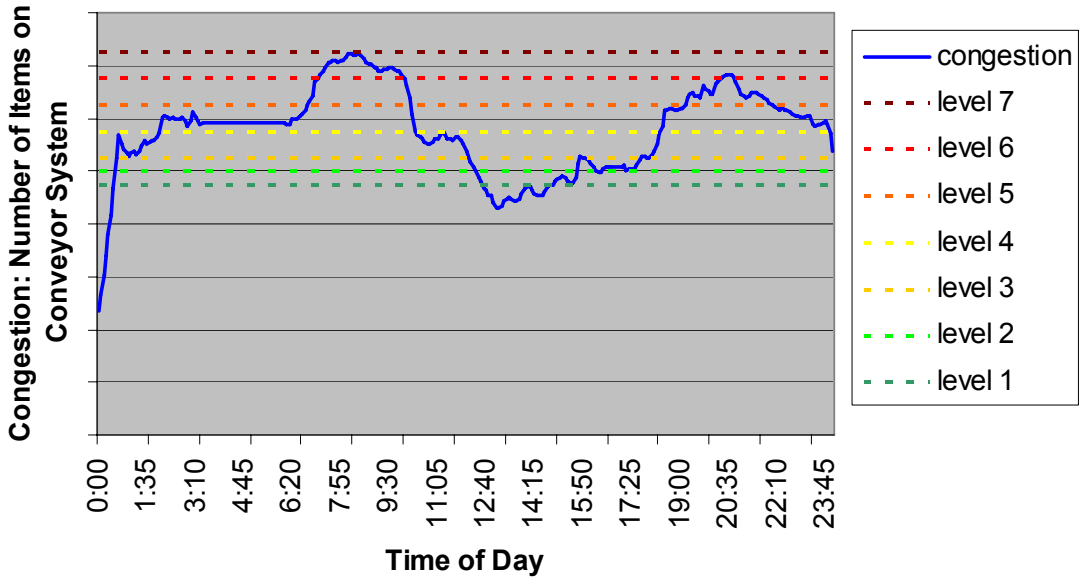


Figure A.3: Number of Items on Conveyor System Over 24 Hours

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

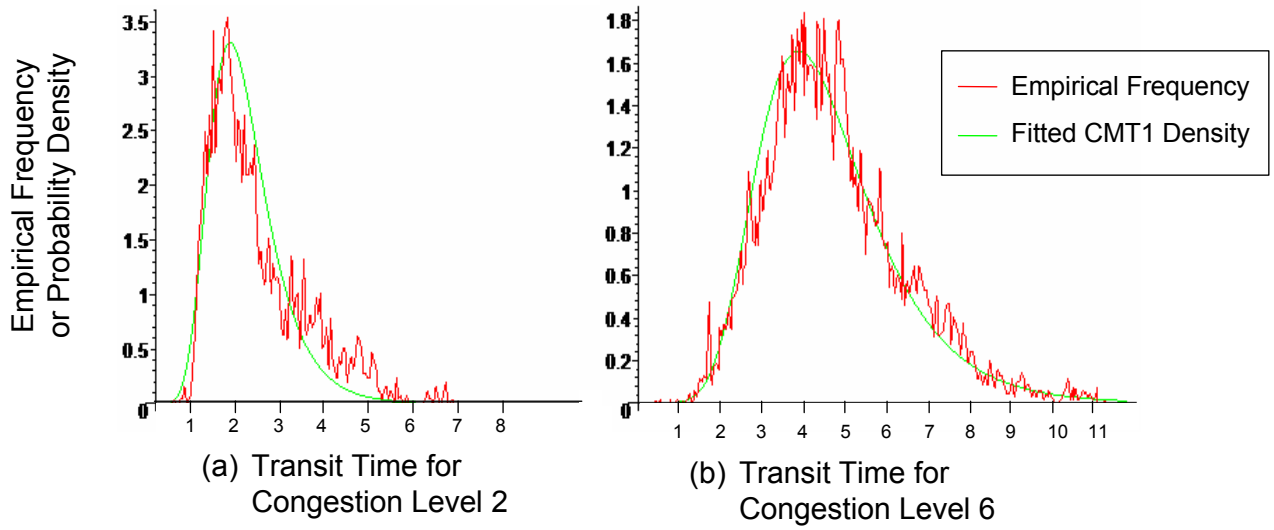


Figure A.4: Empirical Frequency and Fitted $CMT1$ Density for Transit Times from a Conveyor Zone Over 24 Hours for Congestion Levels $g = 2$ and $g = 6$

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

- These empirical transit time distributions seem (mostly) unimodal, seemingly validating at a qualitative level the hypothesis formulated earlier that the heterogeneous behaviors of transit times when observed over long periods of time can be satisfactorily explained by the variations of the system congestion level. Observe that the modes of the distributions represented in Figures A.4 (a) and (b) correspond exactly to the peaks observed on the distribution represented in Figure A.2 around the relative time values 2 and 4.5 respectively.
- The empirical transit time distributions seem to be very well fitted by $CMT1$ distributions. This is remarkable as the $CMT1$ distributions are typically only introduced because of their mathematical properties (as is the case in section §A.6 of this Online Appendix), and not their modeling potential. However, it has already been observed (Gallien and Wein 2001) that $CMT1$ distributions are suitable for modeling transportation times, as their sharp left tail represents typical physical limitations of the transportation means (in the present setting, the conveyor belt speed), while their heavier right tail accounts for all the potential problems encountered along the way (here, congestion at the merge points and delays at the circulation buffer for example).

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

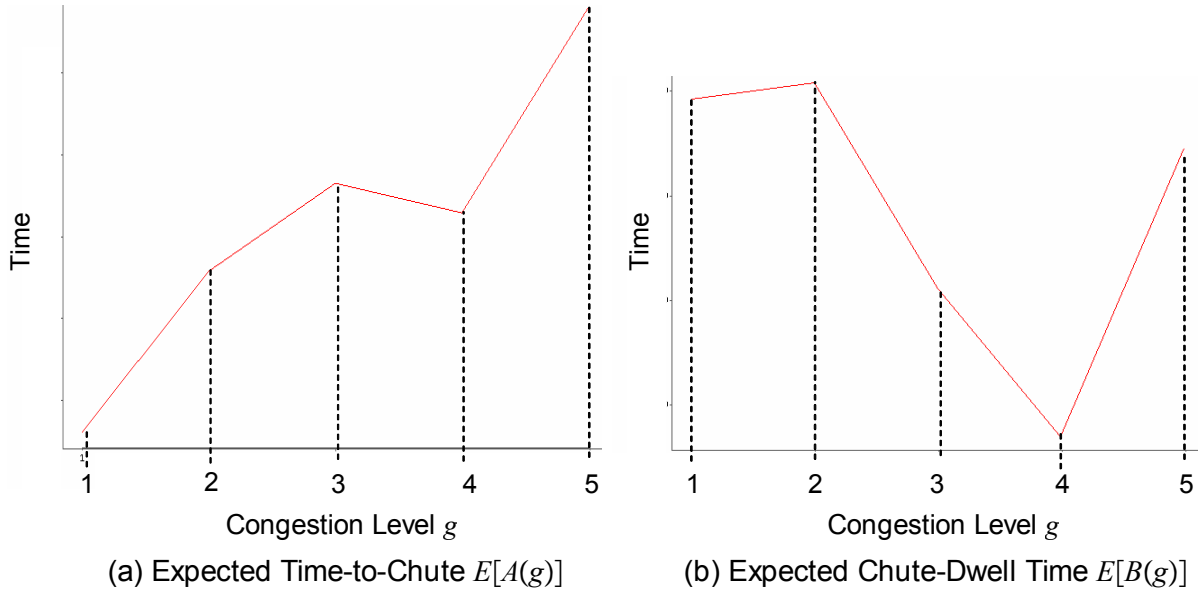


Figure A.5: Variation of Expected Time-to-Chute $E[A(g)]$ and Chute-Dwell Time $E[B(g)]$ with Congestion Level g

observed on several disjoint data sets and with various congestion level definitions, to be explained by the circulating buffer (see §A.1). Specifically, this buffer includes an active tote release logic allowing to dynamically delay the arrival of selected totes to the sorter, with the goal of reducing chute-dwell time for the orders containing items in those totes. The tote delaying logic implemented (which we are not at liberty to describe in more details) does not have any impact for low congestion levels (such as 1 and 2 on Figure A.5 (b)). For medium to high congestion levels (3 and 4 on Figure A.5 (b)) however, the circulating buffer performs its function adequately and the active control logic implemented results in a significant reduction of the average order chute-dwell time. This buffer does have a limited

capacity however, so that when congestion increases further (to 5 on Figure A.5 (b)), it becomes full and, in order to preserve throughput, loses then its ability to increase the sojourn time of selected totes (think of Little's law). The role played by this buffer also explains the slight drop of $\mathbb{E}[A(g)]$ observed in Figure A.5 (a) at congestion level 4, although this is not nearly as significant. In summary, this data analysis uncovered the existence of a non-trivial nominal operating regime (congestion level 4 in Figure A.5) resulting from the design of this process, further motivating the development of order release control policies able to stabilize the process around it (see section §3.5 in the paper).

A.3. Statement and Discussion of Search Algorithm.

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

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

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

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

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

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

feasible for $CDP[\beta]$, and

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

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

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

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

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

A.4. Description of Approximate DP Algorithm.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

A.5. Transient Robustness Experiments.

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

	No Disruption	Experiment 1	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 A.1: Gridlock Probabilities During Transient Simulation Experiments

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

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

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

to gridlock.

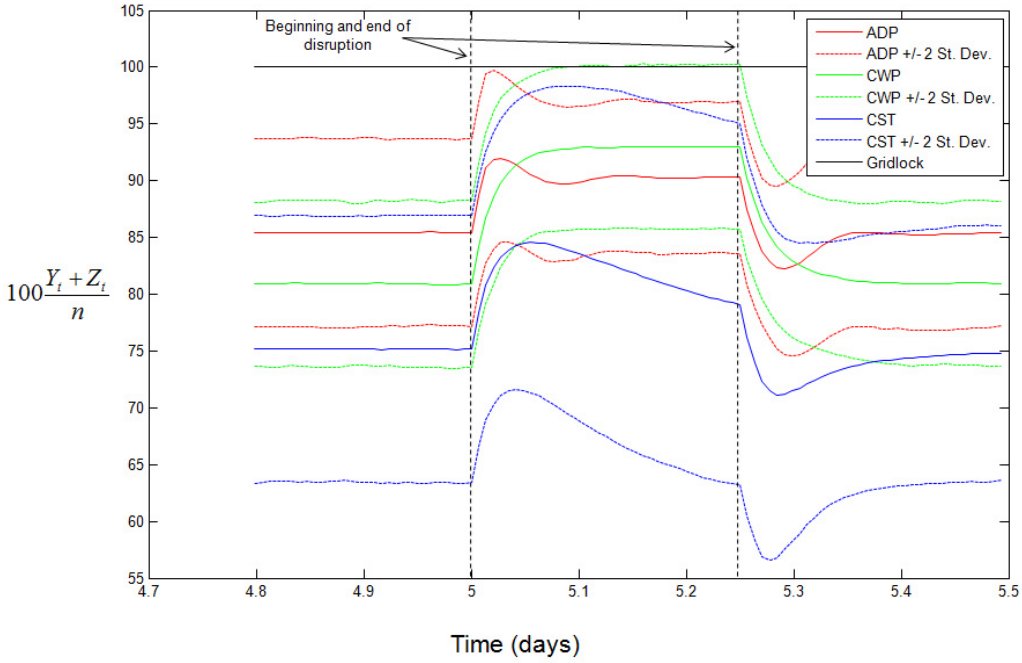


Figure A.6: Evolution of the Average Fraction of Busy Chutes $\frac{Y_t+Z_t}{n}$ during Experiment 1

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

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

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

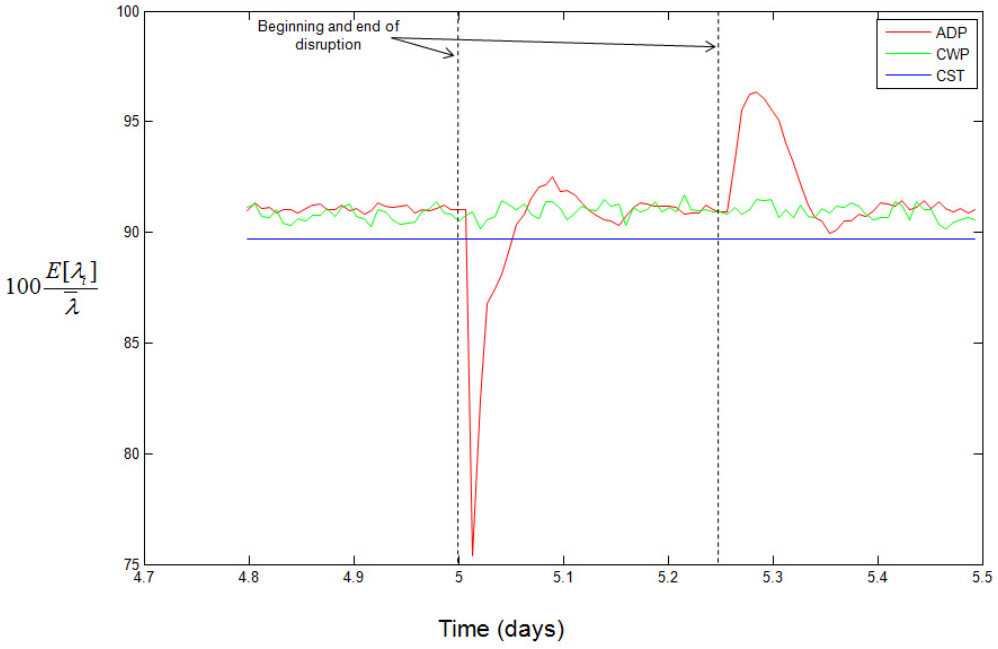


Figure A.7: Evolution of the Average Relative Release Rate $\frac{E[\lambda_t]}{\bar{\lambda}}$ during Experiment 1

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

The transient disruption considered involves an increase of the transition rate of orders between the first and second queues. As result, it creates a simultaneous decrease of the number X_t in the first queue and an increase of the number Y_t in the second, which leaves the total number in system $X_t + Y_t + Z_t$ relatively unaffected. Consequently, the response to that disruption by policy CWP^β is very muted (if observable at all), as seen in Figure A.7. Because CWP^β adjusts its release rate dynamically to keep $X_t + Y_t + Z_t$ at a constant value, but at the same time the sejourn time of orders in the first queue has decreased, the steady state to which the system converges following the immediate transient response to the disruption is one where the number of busy chutes $Y_t + Z_t$ is maintained at a higher value than before (see Figure A.6). For this reason, CWP is the most dangerous policy in that experiment (under CWP^β the system entered gridlock in 49% of replications, under CWP^γ in all of them). Likewise, the release rate of CST^β remains (by definition) exactly identical

throughout the disruption. The initial transient system response under CST^{β} is thus similar to that under CWP^{β}, because for different reasons both policies ignore the initial decrease of X_t and increase of Y_t. However, under CST^{β} the transition rate between the first and second queue starts converging back to the external release rate after an hour or so, causing the number of busy chutes to start decreasing towards its prior steady-state value. For that reason policy CST^{β} fares much better in that experiment than CWP^{β} despite its simplicity, only running the system into gridlock in 5.7% of replications (54% for CST^{γ}, see Table A.1).

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

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

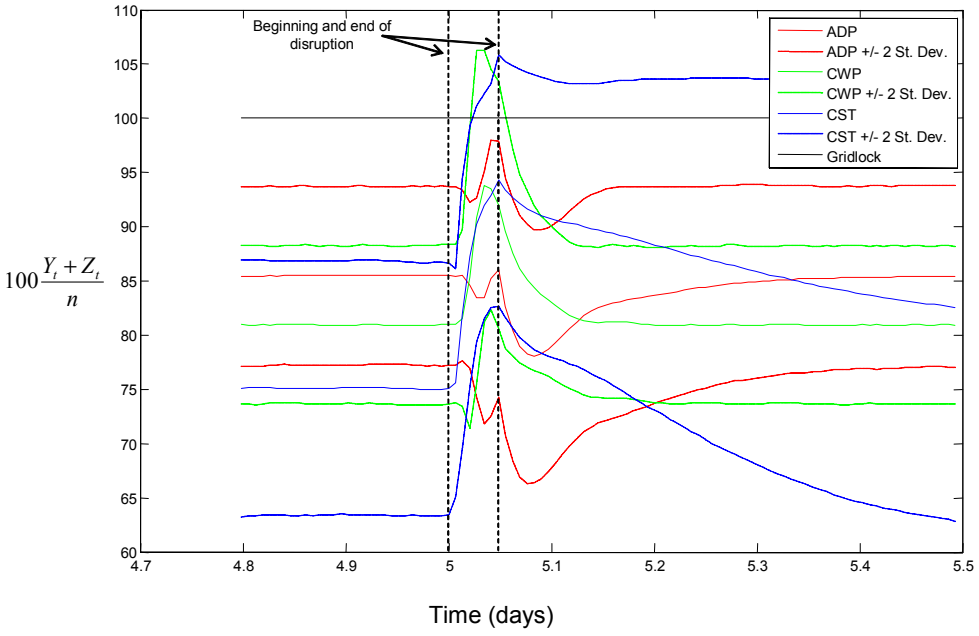


Figure A.8: Evolution of the Average Fraction of Busy Chutes $\frac{Y_t+Z_t}{n}$ during Experiment 2

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

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

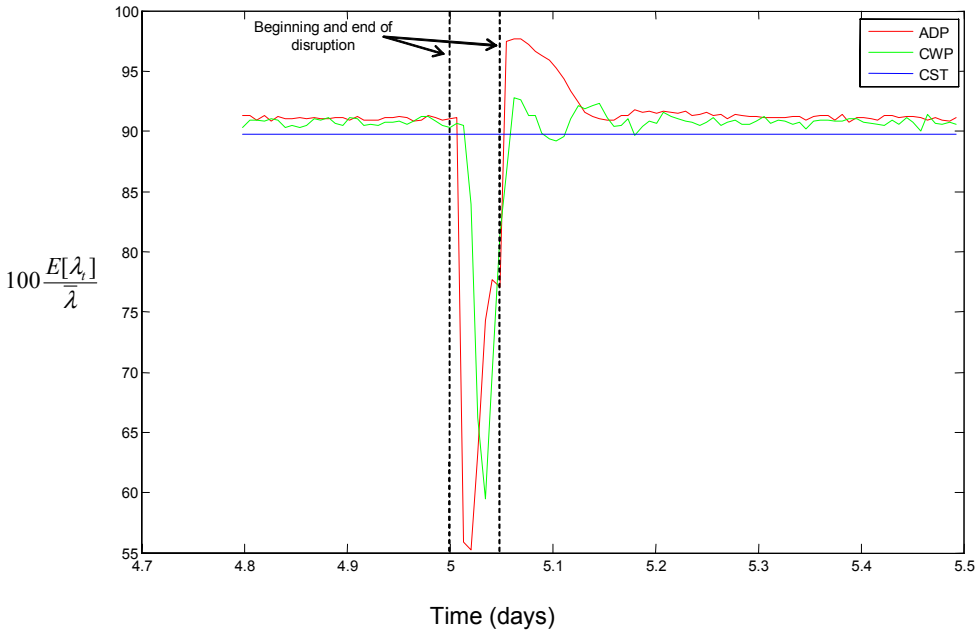


Figure A.9: Evolution of the Average Relative Release Rate $\frac{E[\lambda_t]}{\bar{\lambda}}$ during Experiment 2

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

As seen in Figure A.9, the response by policy ADP to that disruption is qualitatively similar to that of CWP, however ADP responds sooner and with a more drastically reduction of its release rate. This is because ADP is sensitive to the individual value of the number of incomplete chutes Y_t, which immediately starts to increase when the disruption begins. Since the initial queue at the third station is longer under ADP (in simulations $\frac{\mathbb{E}[Z_\infty]}{n} = 19.2\%$ for $w = p$), so is the initial period until the depletion of that queue during which $\frac{dZ_t}{dt} \approx -\frac{dY_t}{dt}$ and

the number of busy chutes remains approximately constant. These features enable policy ADP to overcome the control lag introduced by the service time at the first station (time-to-chute): Figure A.8 shows that the average number of busy chutes under that policy actually decreases after the initial period just described, before the increase in release rate starting around the middle of the disruption period causes it to increase back to about its original value by the end. Because the variability of the number of busy chutes is increased by that disruption, the upper bound of the simulated range for $Y_t + Z_t$ actually increases during the disruption period, and the system under policy ADP did experience gridlock in 0.5% of replications (see Table A.1). That performance is nevertheless substantially better than that of all other policies considered. Also noteworthy is the behavior of ADP after the disruption ends. As Figure A.8 shows, as the disruption ends and the chute-dwell time (service time of the second station) suddenly increases back to its original value, the average number of busy chutes under both CWP and ADP starts to quickly decreases, whereas that reduction and the return to steady-state are considerably slower under CST. However, Figure A.9 shows that, in contrast to CWP, policy ADP is able to exploit that temporary reduction of the number of busy chutes by temporarily increasing its release rate above its original value, that is push more flow into the system (all while maintaining it in a much safer operating regime, as evidenced by Table A.1).

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

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

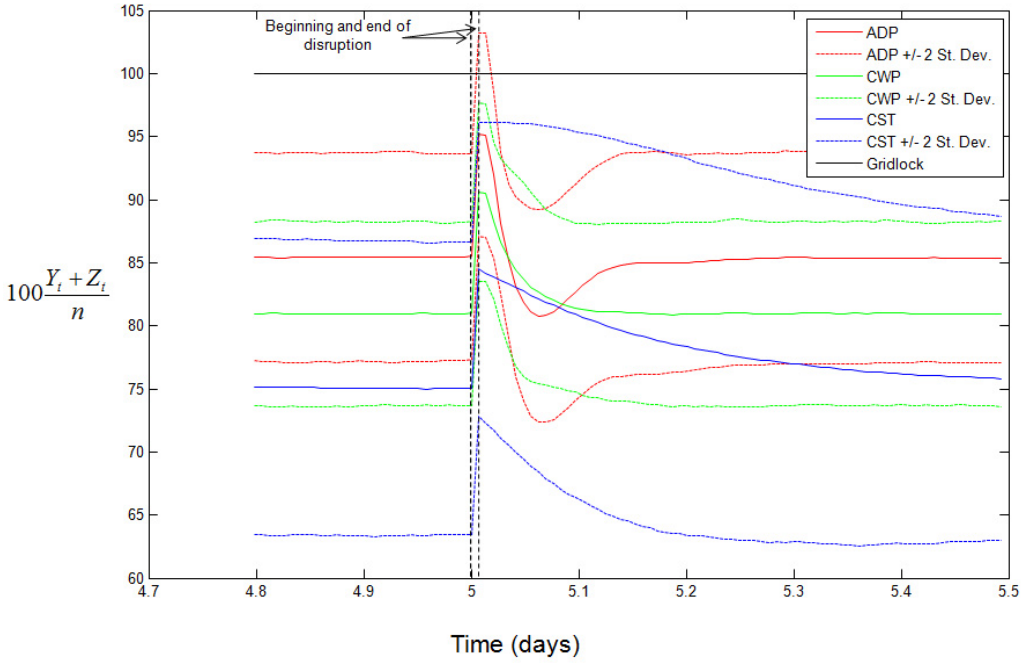


Figure A.10: Evolution of the Average Fraction of Busy Chutes $\frac{Y_t+Z_t}{n}$ during Experiment 3

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

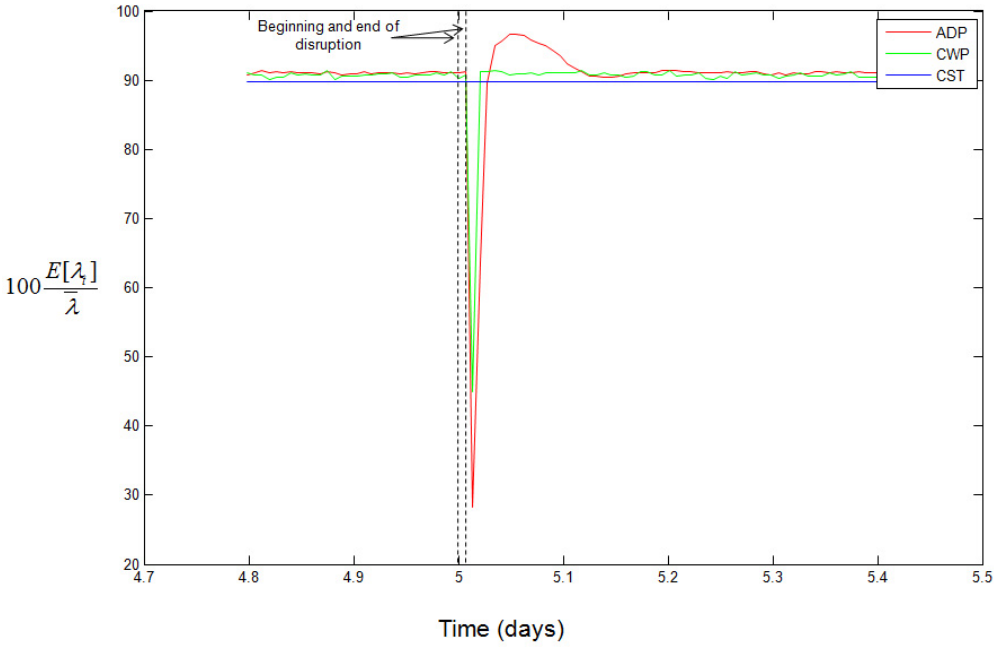


Figure A.11: Evolution of the Average Relative Release Rate $\frac{E[\lambda_t]}{\bar{\lambda}}$ during Experiment 3

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

A.6. Item Release Control Study

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

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

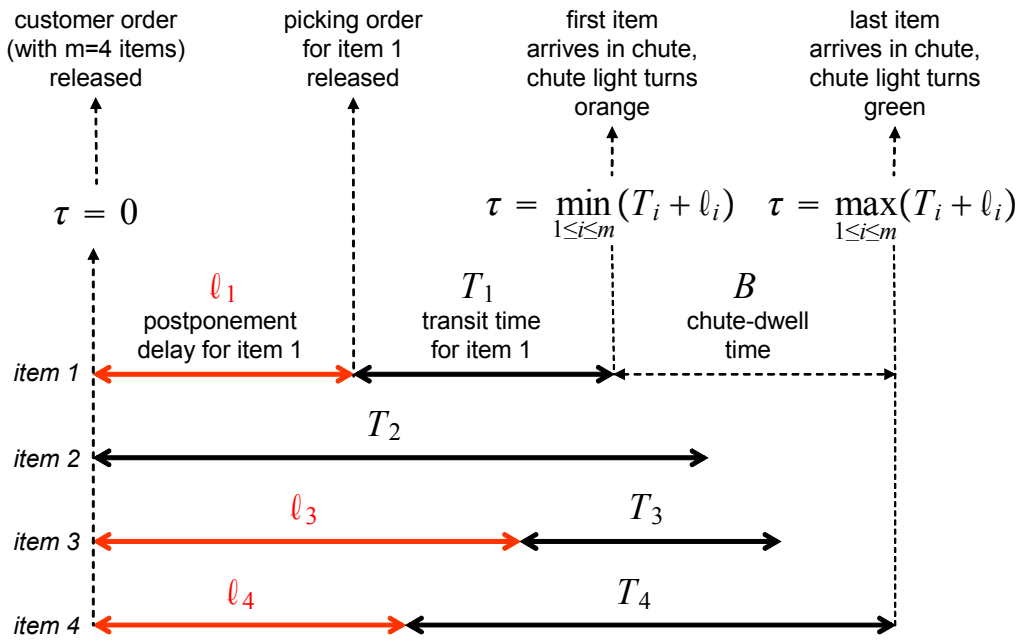


Figure A.12: Timeline of Customer Order through the Pick-to-Ship Process

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

corresponding transfer of any individual items in that order with a positive postponement lead-time (as specified by the control rule just discussed) is then delayed accordingly.

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

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

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

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

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

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

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

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

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

translation we have

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

Also

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

so that

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

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

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

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

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

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

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

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

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

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

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

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

postponement delays given by

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

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

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

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

1. Identifying the current system congestion level g;

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

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

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

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

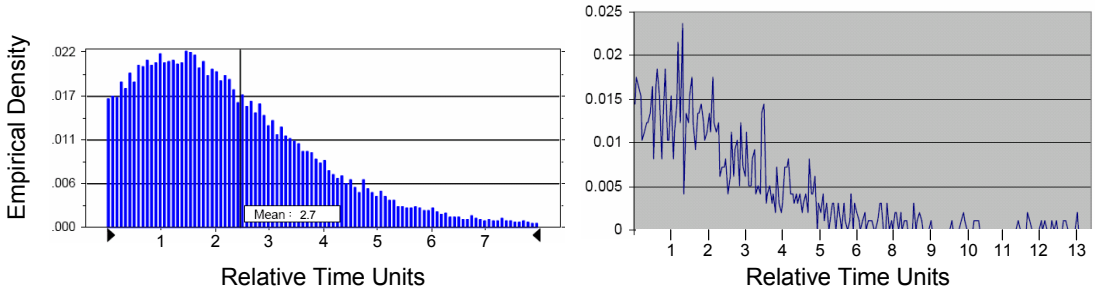
Note that even in the theoretical setting where the item transit time distributions for each order would be independent and exactly given by the $CMT1$ distributions from the set $\mathcal{T}$ defined above, the policy defined by equation (A.11) would likely be suboptimal. This is because the fitted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

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

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

where M (number of items per order) and $\mathbf{R} \triangleq (R_i)_{i \in \{1,...,M\}}$ (congestion zones in which the

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



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

References

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

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

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

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

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

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

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