Dynamic Truck Scheduling using Estimated Times of Arrival

Maximiliano Cubillos
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano 20133, Italy, maximiliano.cubillos@polimi.it

Justin C. Goodson
Richard A. Chaifetz School of Business, Saint Louis University, St. Louis, Missouri 63103, USA, justin.goodson@slu.edu

Ola Jabali
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano 20133, Italy, ola.jabali@polimi.it

Elena Tappia
Dipartimento di Ingegneria Gestionale, via Raffaele Lambruschini 4/B, Milano 20156, Italy, elena.tappia@polimi.it

Abstract. The paper addresses the problem of dynamically scheduling inbound trucks at a warehouse with known service times and uncertain arrival times. Truck arrival time distributions are hidden. However, we approximate them via *estimated times of arrival* (ETAs). Motivated by collaboration with Poste Italiane, which manages one of Italy's largest parcel logistics networks, the objective is to minimize the total expected waiting time. We use information relaxations and an information penalty to develop a dual bound on the cost of an optimal policy. A series of theoretical analyses establishes the dual problem and then transforms it from a stochastic dynamic program to a compact mixed integer linear program. On average, the penalized dual bound is nearly 10 percent stronger than a bound based on perfect information. We propose a lookahead policy that uses ETAs to adapt decisions to new information. When the dispatcher can fully observe truck arrival time distributions, the gap between the policy value and the dual bound is less than one percent. This result suggests that when distributions are hidden, the larger duality gap of about 10 percent we find is due primarily to partial observability and that the policy makes good decisions. Relative to industry practice, the lookahead policy decreases expected waiting time by 29 percent, on average. Further, the lookahead policy selects actions quickly enough to be used in practice.

Key words: truck scheduling, estimated time of arrival (ETA), yard management, POMDP, information relaxation, information penalty

1. Introduction

The global warehousing market was valued at more than one trillion US dollars in 2023 (Grand View Research 2023). Driven by factors including the expansion of e-commerce, increasing globalization, technological advancements, and the growing complexity of logistics chains, the warehousing market is expected to grow more than eight percent annually through 2030 (Grand View Research 2023). In this context, management of inbound trucks at distribution centers and warehouses plays a key role. The short lead times and responsiveness expected by customers depend in part on the scheduling policy that assigns trucks to unloading docks at a warehouse. Upon the arrival of inbound trucks, dispatchers assign them to available docks, otherwise they are assigned to waiting areas. Such assignments are typically made via rules of thumb, e.g., first-come-first-served (Mejía et al. 2023), and rarely account for trucks that will arrive later in the day. These methods often lead to queues, and waiting time incurred by trucks beyond necessary service time can lead to delays that domino across the network. Optimizing truck assignments to account for future arrivals has the potential to reduce waiting time and boost network-wide performance metrics.

Assigning inbound trucks to docks is challenging even when arrival times are known. Indeed, with only a single dock, the problem is strongly NP-hard (Pinedo 2022). In practice, truck arrival times are uncertain for many reasons. In most transportation research, uncertain arrival times are characterized by probability distributions. It is typically assumed that these distributions are known or can be estimated from data. However, this assumption is impractical for inbound truck applications where multiple carriers operate across various logistics networks. In such cases, a carrier separately manages routes for its vehicles, and a visit to a particular warehouse is only one of many stops. Daily adjustments by carriers to origins, destinations, and routes—which are unknown to the warehouse dispatcher—make it difficult to meaningfully characterize probability distributions on truck arrival times. Moreover, even if the dispatcher was privy to carriers' plans, she does not have the means to collect the data necessary to estimate the distributions. For each possible route, estimation would require repeated observation of arrival times that vary dynamically due to congestion, service times at other stops, breaks for drivers, weather, and vehicle failures. Thus, it is is unrealistic to suppose that a warehouse can collect such data.

Modern sensing technology can mitigate the challenges surrounding the estimation of truck arrival times. Today, many carriers outfit their trucks with devices that provide real-time locations. With help from third-party software firms, the devices facilitate point estimates of arrival times. These *estimated times of arrival* (ETAs) are communicated with high frequency to the carriers, and we assume the carriers share ETAs with the warehouse. The ETAs may be treated as noisy signals of arrival time distributions the warehouse cannot directly observe. Through filtering, ETAs enable rolling approximations of truck arrival time distributions, which we call *belief distributions*. In this paper, we show that belief distributions facilitate scheduling policies for inbound trucks that significantly reduce waiting times relative to current practice.

A conventional approach to dynamically scheduling inbound trucks models the problem as a Markov decision process (MDP), which assumes truck arrival time distributions are readily available. However, because in reality the distributions are hidden from the dispatcher, an MDP is difficult to operationalize. Instead, we formulate the problem as a partially-observable Markov decision process (POMDP). In contrast to an MDP model, our POMDP model uses ETAs to update belief distributions on truck arrival times. Although the addition of belief distributions to the state variable makes the model more complex, a POMDP more accurately models the actual information available to dispatchers.

Our POMDP model is motivated by collaboration with the logistics division of Poste Italiane, which manages one of Italy's largest parcel logistics networks. As Poste Italiane looks to improve network-wide lead times and responsiveness, it is leveraging ETAs to upgrade scheduling at its warehouses. Although Poste Italiane's warehouses have many docks, they are partitioned into operational groups according to features, such as unloading equipment, product category, and storage constraints. Before their arrival to the warehouse, inbound trucks are tied to an operational group. Because each group functions separately, trucks in one group may be scheduled independently from trucks in another group. Thus, we focus on scheduling trucks across a single group. Upon a truck's arrival, the dispatcher assigns the truck to an available dock within its group or directs the truck to wait in the yard. The dispatcher may direct the truck to wait even if one or more docks are available. Because docks within a group are similar, they may be treated as a pool of identical resources. Thus, a truck may be assigned to any dock within the group and the truck's service time, which we assume is known, is the same across all docks in the group. Once service begins, it cannot be preempted. Poste Italiane's objective is to minimize the total expected waiting time across all trucks over the operating horizon.

We propose an approximate algorithm to dynamically schedule trucks. Our policy utilizes a lookahead mechanism to estimate the future costs of a decision taken from a given state. The lookahead mechanism takes as input a scenario of truck arrival times, then uses iterated local search (ILS) to approximately solve the corresponding deterministic scheduling problem. A decision is evaluated by employing the lookahead mechanism across a range of scenarios that may be encountered one step into the future. As belief distributions evolve with the receipt of ETAs, so do the likelihoods of the scenarios. In this way, the policy adapts its decisions to new information.

The vast majority of dynamic and stochastic transportation research studies assess solution quality by comparison to benchmark policies. This approach highlights relative improvement but leaves open the question of how much better an optimal policy might be. Among studies that establish a dual bound on the cost of an optimal policy, most work with perfect information relaxations of an

MDP model. This is a natural approach because the perfect information problem often corresponds to a familiar deterministic problem. However, because the value of information is high, this method often results in a loose bound.

We use two information relaxations and an information penalty to develop a dual bound that is much stronger than a bound obtained only through perfect information. In addition to a perfect information filtration, we employ a *distribution* filtration that gives the dispatcher knowledge of the unobserved truck arrival time distributions. As we show, scheduling under the distribution filtration is equivalent to optimization via an MDP model. Our dual bound allows trucks to be scheduled with perfect knowledge of future arrival times, but it penalizes the dispatcher for using this information. Specifically, instead of accruing known wait times, the dispatcher accrues wait times in expectation. Expected waiting time is calculated via MDP contributions. In this way, the dual problem incentivizes a clairvoyant dispatcher to select actions as if she were not wholly aware of actual truck arrival times. Penalizing decisions in this fashion not only yields a dual bound for the POMDP, but also for the MDP. A series of theoretical analyses establishes the dual problem and then transforms it from a stochastic dynamic program to a compact mixed integer linear program (MILP). The mathematical program representation is desirable because the MILP is amenable to solution via commercial solvers. Although our analyses are specific to inbound truck scheduling with ETAs, they serve as a template for the transportation science and logistics community to develop dual bounds for other problems.

The lookahead policy and dual bound are vetted on a range of problem instances representative of real operating scenarios. Smaller instances correspond to settings often found in urban operations or retail receiving, which typically manage a handful of trucks across one or two docks during a given day. The scale of these instances is modest enough to facilitate comparison of the lookahead policy with the dual bound, which eventually becomes intractable as problem size grows. Because the dual bound is valid for both the POMDP and MDP settings, we gauge policy quality via both models. Experiments show that the dual bound is much stronger than a more conventional bound based on perfect information. On average, the penalized dual bound is nearly 10 percent larger. Moreover, the bound approaches its upper limit. When the lookahead policy is applied to the MDP, where truck arrival time distributions are given, the average duality gap is less than one half of one percent. Thus, we show empirically that for practical purposes the lookahead policy is nearly an optimal MDP policy. When the lookahead policy is implemented in the more realistic POMDP setting, where truck arrival time distributions are hidden, the duality gap is larger. However, the lookahead policy's strong performance in the MDP setting suggests that the larger gap is primarily due to partial observability rather than suboptimality. Because the policy must work with belief distributions on arrival times instead of with actual distributions, policy cost is higher and the duality gap is bigger. Larger instances are inspired by our collaboration with Poste Italiane. These require the management of many truck arrivals across many docks. Experiments on larger instances demonstrate significant reductions in waiting relative to a first-come-first-served policy, which is common in practice (Mejía et al. 2023). On average, the lookahead policy decreases expected waiting time by 29 percent. Experiments also show that the lookahead policy selects actions quickly enough to be used in real time.

Additional computational work points to the utility of ETAs. The extant literature on truck scheduling treats ETAs as point forecasts for actual arrival times. Experiments show that filtering ETAs to maintain belief distributions on truck arrival times leads to substantially better scheduling decisions. Without filtering, the lookahead policy incurs waiting time as if it does not look ahead at all. With expected waiting time nearly nine percent higher, on average, without filtering the lookahead policy performs like the rolling horizon procedures that dominate the truck scheduling literature. The ability to improve decisions hinges on using ETAs to characterize uncertainty in truck arrival times. We also gauge the usefulness of ETAs as their volatility increases. Even when ETAs are relatively weak signals of unobserved arrival time distributions, they still facilitate lookahead policies that handily outperform industry practice.

To summarize, our contributions are as follows:

- **Dynamic Scheduling Model with ETAs.** We formulate a POMDP model to dynamically schedule inbound trucks to a warehouse in the face of uncertain arrival times. In contrast to much of the research in transportation, our model recognizes that dispatchers do not have direct access to distributions. Our model filters ETAs to maintain belief distributions on truck arrival times.
- **Penalized Dual Bound.** We develop a dual bound on the cost of an optimal policy. In contrast to most transportation research, which gauges quality via comparison to benchmarks or to loose bounds based on perfect information, we use an information penalty to obtain a much stronger bound.
- **Lookahead Policy.** We propose a lookahead policy that uses ETAs and belief distributions on truck arrival times to adapt its decision making. When the dispatcher can fully observe truck arrival time distributions, the duality gap is very small, indicating that the policy is nearly

optimal. This result suggests that when distributions are hidden, a larger duality gap is due primarily to partial observability. Additionally, the policy is suitable for real-time decisions, even at scale.

- **Value of ETAs.** The literature on truck scheduling treats ETAs as point estimates. We show that filtering ETAs to characterize uncertainty in truck arrival times is key to better scheduling decisions. We also show that even when ETAs are volatile signals of truck arrival time distributions, they lead to policies that outperform industry practice.

The paper proceeds as follows. We review related literature in §2. The POMDP model is presented in §3. The loookahead policy is described in §4. The dual bound is developed in §5. Computational experiments are outlined in §6. We present conclusions in §7.

2. Related Literature

Our research contributes to the literature on truck scheduling with uncertain arrival times and machine scheduling with random releases. In this section, we discuss how our work is similar to and different from the research in these areas.

The literature on truck scheduling with uncertain arrival times consists of static and dynamic solution methodologies. Static approaches fix a schedule at the beginning of the operating horizon and follow it no matter when trucks arrive. In the context of cross-dock operations, Konur and Golias (2013) and Heidari et al. (2018) frame such problems as a bi-level program that minimizes waiting and service costs. They obtain heuristic solutions via a genetic algorithm. Xi et al. (2020) optimally solve a robust formulation. In each study, the goal is to identify a schedule that performs well across a range of arrival times.

Dynamic solution methods aim to improve on static methods by allowing the schedule to change in response to realized truck arrival times. Yu et al. (2008) treat the problem of minimizing the expected labor time required to process an uncertain volume of freight at a consolidation facility. When a truck arrives, it is assigned to the dock that minimizes its immediate service time. Song et al. (2022) consider both routing and scheduling costs. In the second stage of their two-stage stochastic program, scheduling is governed by a first-come-first-served policy, meaning trucks queue for docks based on the order in which they arrive. While neither method operates based on a fixed schedule, both are myopic in the sense that there is no consideration of how actions in the present are connected to subsequent decisions.

In contrast, rolling horizon methods are more anticipatory. When these methods assign trucks to docks, they consider the potential impact of a decision now and across future periods. This is typically accomplished via optimization across upcoming assignments, with random arrival times replaced by mean values. The objective of the rolling horizon's deterministic optimization reflects the objective of the underlying dynamic and stochastic problem. In each of the following references, the rolling horizon objective is to minimize total wait time or wait time due to deviation from an initial schedule. Cekała et al. (2015) and Xu et al. (2022) use genetic algorithms to heuristically solve the rolling horizon problem. Larbi et al. (2011) employ a greedy heuristic. Nasiri et al. (2022) use a solver to obtain exact solutions. The methods of Vanga et al. (2022) and Modica et al. (2024) take current ETAs as truck arrival times instead of mean values. Modica et al. (2024) heuristically solve the rolling horizon problem via a genetic algorithm and Vanga et al. (2022) use a solver to find exact solutions. Whether solutions to the rolling horizon optimization are heuristic or exact, the mechanism is an approximation of the original problem and carries no performance guarantee.

Our research draws on aspects of both static and dynamic methodologies for truck scheduling. Although our lookahead policy is dynamic, it uses static schedules to select actions. But unlike the rolling horizon approach, which operates on the current state, our lookahead policy considers how decisions might be made across a range of future scenarios. In contrast to the dynamic truck scheduling literature, which fails to rigorously model state dynamics and uncertainties, our POMDP model recognizes that ETAs can change across time and that beliefs about arrival times should consequently be modified. Furthermore, we leverage relaxations of the POMDP to establish a dual bound. Until now, the dynamic truck scheduling literature has relied on comparison to heuristic policies to gauge quality. Because our dual bound is an absolute benchmark, it provides a means of measuring quality relative to the cost of an optimal policy. Moreover, the rigorous incorporation of ETAs into model, method, and dual bound adds a realistic dimension to our work that is not found in the extant literature.

Truck scheduling with uncertain arrival times is analogous to machine scheduling with stochastic release dates: jobs correspond to trucks, machines correspond to docks, and arrival times correspond to release dates. Whereas truck scheduling usually seeks to minimize waiting time, machine scheduling objectives vary. They include the minimization of makespan, completion time, and weighted tardiness. To emphasize the difficulty of this class of problems, even when release dates are known with certainty and there is only one machine, the problem is strongly NP-hard (Pinedo 2022).

Like the truck scheduling literature with uncertain arrival times, the machine scheduling literature with stochastic release dates is limited. Liu et al. (2021) approach the problem in two stages.

First, assign jobs to machines. Then, after uncertainties are realized, jobs are sequenced on their assigned machines. The objective is to minimize setup costs plus penalties for early and late completions. Zhang et al. (2012) and Heydar et al. (2022) propose fully dynamic reinforcement learning methodologies. When a job is released, assignment to a machine is determined by a linear approximation of the cost-to-go. Estimates draw on state features such as waiting jobs, idle machines, and remaining processing time. Zhang et al. (2012) search for policies that minimize weighted tardiness, while Heydar et al. (2022) seek energy efficient policies that minimize makespan. The linear cost-to-go estimate of Ronconi and Powell (2010) uses state-dependent static schedules to learn parameter values in pursuit of policies that minimize total tardiness.

Our work is applicable to machine scheduling problems with stochastic release dates, with known processing times, with identical machines, and without preemption. In this setting, our dual bound is a notable contribution. As we demonstrate, with full knowledge of release/arrival time distributions, the gap between our lookahead policy and the dual bound is nearly zero, implying that the policy is very good. Further, because the literature does not contain an analog to the ETA, such as an estimated time of release, we address a more general problem.

Finally, the machine scheduling literature also considers the case of online release dates, meaning release dates are random, but no distributional information is available. In contrast to the stochastic case, the literature surrounding online release dates provides absolute benchmarks. For example, Megow et al. (2006) and Gupta et al. (2020) establish performance guarantees, and Chou et al. (2006) show that a list-style policy is optimal as the number of jobs tends to infinity. Although uncertain release dates connect this stream of research to our own, the problem we address is fundamentally different. While online releases may be common in machine scheduling, truck arrivals to a warehouse are typically anticipated. Consequently, probabilistic information plays a central role in our model, method, and dual bound.

3. Model

Problem Description and POMDP. We consider the problem of assigning inbound trucks to loading docks at a warehouse. Each truck's time of arrival to the warehouse is unknown and the distribution of possible arrival times is unobserved. Across the operating horizon, the decision maker receives estimated times of arrival (ETAs) at known frequencies. The decision maker uses ETAs to adjust belief distributions on truck arrival times. Each belief distribution characterizes the uncertainty in arrival time. Upon arrival to the warehouse, each truck is either assigned to an available dock or waits at the yard for future assignment. A truck occupies its assigned dock for the duration of its known service time. Service is non-preemptive. The objective is to minimize the expected total waiting time of the trucks. We formulate the problem as a partially-observed, semi-Markov decision process (POMDP). The model consists of an information process that connects truck arrival times to ETAs alongside a decision process that responds to belief distributions on arrival times. Both are described in what follows.

ETAs and Belief Distributions. Each truck's ETAs and arrival time distribution follow a hidden Markov model, meaning the arrival time distribution is unobserved, but can be estimated from observed ETAs. Because the decision maker does not know the relationship between ETAs and the arrival time distribution, she employs Bayesian filtering (Särkkä 2013) to update belief distributions for each truck. Denote the set of trucks by $\mathcal{J} = \{1, \ldots, J\}$ and the start of the operating horizon by t_0. Let $e_j(t)$ be the set of ETAs received for truck $j \in \mathcal{J}$ to and including time t, ordered as they are received, where each ETA is in the range $[t_0, \infty)$. Let A_j be the random variable describing the arrival time of truck $j \in \mathcal{J}$ with support $[t_0, \infty)$ and unobserved distribution F_{A_j}. Let $F_{A_j}(t)$ be the belief distribution on the arrival time for truck j at time t. If truck j has not yet arrived by time t, then its belief distribution requires realizations to occur at later times, meaning $F_{A_j}(t)$ is conditional on $A_j > t$. Belief distributions inform state transition probabilities in the decision process. They are updated whenever ETAs are received. The initial belief distribution $F_{A_j}(t_0)$ may be taken from historical data. We require the belief distribution to converge to the unobserved distribution by the arrival time. Denoting by a_j a realization of A_j, this requires $F_{A_j}(a_j) = F_{A_j}$.

Decision Epochs. Decision epochs are indexed by $k = 0, \ldots, K$. An epoch is triggered by any of four events: (i) *Arrivals.* Arrival of a truck to the warehouse. We assume trucks arrive one at a time. (ii) *Service Completions.* Completion of service for one or more trucks assigned to docks. (iii) *ETAs.* Receipt of an ETA when at least one truck is waiting and at least one dock is available. This event allows the decision maker to pivot in response to new arrival time estimates when resources are available and trucks are queued. Although the receipt of every ETA updates a belief distribution, an ETA only triggers a decision epoch under these circumstances. (iv) *Trucks at the Yard and Available Docks.* Assignment of a truck to a dock when at least one truck is in the yard and at least one dock is available. This event facilitates the assignment of multiple trucks to docks in consecutive epochs occurring at the same time.

States. The state s_k of the process at epoch k includes the current time, a history of ETAs, truck arrival times as they are revealed, belief distributions on arrival times, and service start times.

Let $t_k \in [t_0, \infty)$ be the time at which epoch k occurs. Prior to truck j's arrival, its time of arrival $a_j = ?$ is unknown. Let $u_j \in [t_0, \infty) \cup \{?\}$ be the time at which truck $j \in \mathcal{J}$ begins service, where $u_j = ?$ until a start time is determined. Then, the state is $s_k = (t_k, (e_j(t_k), a_j, F_{A_j}(t_k), u_j)_{j \in \mathcal{J}})$. In initial state $s_0 = (t_0, (e_j(t_0)), ?, F_{A_j}(t_0), ?)_{j \in \mathcal{J}})$, the current time is t_0, an initial set of ETAs $e_j(t_0)$ is available for each truck, the decision maker has an initial belief distribution $F_{A_j}(t_0)$ for all $j \in \mathcal{J}$, and all trucks are en route to the warehouse. A terminal state $s_K = (t_K, (e_j(t_K), a_j, F_{A_j}, u_j)_{j \in \mathcal{J}})$ is any state such that $a_j \neq ?$ and $u_j \neq ?$ for all $j \in \mathcal{J}$, meaning all trucks have arrived and entered service.

Actions. An action is an assignment of a truck to a dock. Trucks eligible for assignment include trucks waiting at the warehouse and trucks that have just arrived at the warehouse. Denote the set of docks by $\mathcal{D} = \{1, \ldots, D\}$ and the service time for truck $j \in \mathcal{J}$ by p_j. For convenience, let $\mathcal{J}^d(s_k) = \{j \in \mathcal{J} : a_j = ?\}$ be the subset of trucks still en route to the warehouse in state s_k, let $\mathcal{J}^w(s_k) = \{j \in \mathcal{J} : a_j \neq ?, u_j = ?\}$ be the subset of trucks waiting at the warehouse for assignment to a dock in state s_k, let $\mathcal{J}^p(s_k) = \{j \in \mathcal{J} : u_j \neq ?, u_j \leq t_k \leq u_j + p_j\}$ be the subset of trucks in process in state s_k, and let $\bar{D}_k(s_k) = D - |\mathcal{J}^p(s_k)|$ be the number of available docks in state s_k. Let $x_{kj} = 1$ if truck $j \in \mathcal{J}^w(s_k)$ is assigned to a dock at epoch k and 0 otherwise. When the process occupies state s_k, the set of feasible actions is $\mathcal{X}(s_k) = \{x_k \in \{0,1\}^{|\mathcal{J}^w(s_k)|} : \sum_{j \in \mathcal{J}^w(s_k)} x_{kj} \leq \min\{1, \bar{D}_k(s_k)\}\}$. The constraint that defines the set allows at most one truck in $\mathcal{J}^w(s_k)$ to be assigned to an available dock. If all docks are unavailable, then no assignment can be made. Setting an element of x_k to 0 directs the corresponding truck to wait. Managing truck assignments this way, in one-at-a-time fashion, reflects industry practice.

Transitions. Following action selection, the transition to the post-decision state sets service start time to $u_j = t_k$ for the truck $j \in \mathcal{J}^w(s_k)$ such that $x_{kj} = 1$, if such a truck exists. All other state components remain the same. Denote the post-decision state by s_k^x. The next decision epoch is marked by a transition to state s_{k+1} and occurs at random time t_{k+1}. Time t_{k+1} is the time of the event that triggers the epoch. We express t_{k+1} as the earliest of the four trigger events outlined above as follows: (i) *Arrivals.* Let $\bar{A}_k = \min\{A_j : j \in \mathcal{J}^d(s_k^x)\}$ be the random variable describing the earliest arrival time of trucks en route to the warehouse. If $\mathcal{J}^d(s_k^x)$ is empty, then we set $\bar{A}_k$ to ∞. (ii) *Completions.* Let $\bar{p}_k = \min\{u_j + p_j : j \in \mathcal{J}^p(s_k^x)\}$ be the earliest time at which a truck finishes service. If $\mathcal{J}^p(s_k^x)$ is empty, then we set $\bar{p}_k$ to ∞. (iii) *ETAs.* Let $\bar{e}_k$ be the time of the next ETA after time t_k. If the set of waiting trucks $\mathcal{J}^w(s_k^x)$ is empty, or if the number of available docks $\bar{D}_k(s_k^x)$ is zero, then we set $\bar{e}_k$ to ∞. (iv) *Trucks at the Yard and Available Docks.* Let $\bar{t}_k = t_k$ be the current time. If action x_k makes no assignment, if the set of waiting trucks $\mathcal{J}^w(s_k^x)$ is empty, or if the number of available docks $\bar{D}_k(s_k^x)$ is zero, then we set $\bar{t}_k$ to ∞. Then, $t_{k+1} = \min\{\bar{A}_k, \bar{p}_k, \bar{e}_k, \bar{t}_k\}$.

Because arrival time distributions are unobserved, belief distributions characterize the uncertainty surrounding $\bar{A}_k$, and thus the uncertainty surrounding t_{k+1}. The transition from post-decision state s_k^x to pre-decision state s_{k+1} sets the time of epoch $k+1$ to the observed realization of t_{k+1}. If one or more ETAs are received for truck j between times t_k and t_{k+1}, or if an ETA triggers epoch $k+1$, the set $e_j(t_{k+1})$ is obtained by adding these ETAs to $e_j(t_k)$. Otherwise, $e_j(t_{k+1})$ is equal to $e_j(t_k)$. Belief distributions are updated for each ETA receipt as previously described. If epoch $k+1$ is triggered by the arrival of truck j, then $a_j = t_{k+1}$.

Contributions. The contribution at epoch k is the expected waiting time incurred until the next epoch. For trucks that have arrived at the warehouse, but whose service has not yet started, this waiting time is the expected duration of period k, i.e., the expected time between t_k and t_{k+1}. Let

$$W_{kj}(s_k, x_k, t_{k+1}) = \begin{cases} t_{k+1} - t_k, \ j \in \mathcal{J}^w(s_k) \text{ and } x_{kj} = 0, \\ 0, \ \text{otherwise}, \end{cases} \tag{1}$$

be the random waiting time incurred by truck j when the process occupies state s_k, action x_k is selected, and the next epoch begins at random time t_{k+1}. A truck waits for the duration of period k if it belongs to $\mathcal{J}^w(s_k)$ in state s_k and action x_k does not assign the truck to an available dock. The contribution $W_k(s_k, x_k) = \mathbb{E}[\sum_{j\in\mathcal{J}} W_{kj}(s_k, x_k, t_{k+1})|s_k, x_k]$ is the expected total waiting time across trucks conditional on s_k and x_k, where the expectation is taken with respect to random time t_{k+1}.

Objective. A policy $\pi = (X_0^\pi, \ldots, X_K^\pi)$ is a sequence of decision rules where each rule $X_k^\pi(s_k)$: $s_k \rightarrow \mathcal{X}(s_k)$ maps the current state to a feasible action. The cost $W^\pi = \mathbb{E}[\sum_{k=0}^K W_k(s_k, X_k^\pi(s_k))|s_0]$ of policy π is the expected sum of contributions conditional on initial state s_0. Expectation is taken with respect to state trajectories. Denote by Π the set of all policies. The objective is to identify a policy in Π with minimal cost: $W^\star = \min\{W^\pi : \pi \in \Pi\}$.

4. Lookahead Policy

We use a one-step lookahead policy to select actions. The lookahead mechanism solves a deterministic version of the inbound truck scheduling problem. We use iterated local search (ILS) to identify a heuristic solution to the problem. We describe the lookahead mechanism in §4.1, the policy in §4.2, and procedures to accelerate computation in §4.3.

4.1. Lookahead

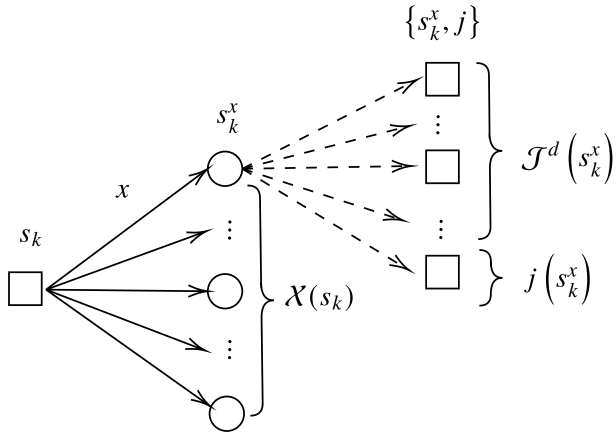
The lookahead mechanism operates on a given state s and a given vector of arrival times $\hat{a} = (\hat{a}_j)_{j \in \mathcal{J}}$. For each truck $j \in \mathcal{J}^d(s)$ en route to the warehouse, the vector specifies an arrival time $\hat{a}_j$. Otherwise, $\hat{a}_j = a_j$ is the arrival time given in state s. Let γ be an ordering of the trucks in $\mathcal{J}^d(s) \cup \mathcal{J}^w(s)$ that are either en route to the warehouse or waiting at the warehouse for assignment to a dock. Trucks are assigned to docks as soon as possible in the sequence specified by γ and per the arrival times $\hat{a}$. Denote the g-th truck in the order by $\gamma(g)$. The service start time $u_{\gamma(g)}$ for the g-th truck is the latest of the current time, the arrival time of the g-th truck, the service start time of the previous truck in the order, and the earliest time at which a dock is available. Denote by $c_j = u_j + p_j$ the service completion time for truck j. Denote by $t(g-1)$ the earliest time at which one of the D docks becomes available after the first $g-1$ trucks in γ are serviced. Letting $\max_{(D)}\{\cdot\}$ be the D-th largest element of a set, then $t(g-1) = \max_{(D)}\{\{c_j : j \in \mathcal{J}^p(s)\} \cup \{c_{\gamma(1)}, \ldots, c_{\gamma(g-1)}\}\}$ is the D-th largest service completion time among trucks in process in state s plus the first $g-1$ trucks in order γ. If the number of such trucks is less than D, then take $t(g-1) = t$ to be the time at which the process occupies state s. Additionally, take $t(0) = \max_{(D)}\{c_j : j \in \mathcal{J}^p(s)\}$. Then, the service start time for the g-th truck in order γ is $u_{\gamma(g)} = \max\{t, \hat{a}_{\gamma(g)}, u_{\gamma(g-1)}, t(g-1)\}$.

Let $\hat{W}(s, \hat{a}, \gamma) = \sum_{g=1}^{|\gamma|} u_{\gamma(g)} - \hat{a}_{\gamma(g)}$ be the waiting time associated with a state s, arrival times $\hat{a}$, and an order γ. Denote by $\Gamma(s)$ the set of all permutations of trucks in $\mathcal{J}^d(s) \cup \mathcal{J}^w(s)$. We use ILS (Lourenço et al. 2003) to heuristically solve the optimization problem $\min\{\hat{W}(s, \hat{a}, \gamma) : \gamma \in \Gamma(s)\}$. The ILS procedure alternates between a perturbation phase and a local search phase. The perturbation phase diversifies the search by randomly relocating a portion of the trucks in an order. The local search phase intensifies the search via a first-improving search across four neighborhood structures: two-way swap, insertion, four-way swap, and insertion plus two-way swap. These neighborhoods are similar to those explored in Maecker et al. (2023) for unrelated parallel machine scheduling. If the ILS procedure returns an order $\hat{\gamma}$ in state s with arrival times $\hat{a}$, then denote the corresponding objective value by $\hat{W}(s, \hat{a}) = \hat{W}(s, \hat{a}, \hat{\gamma})$. The Appendix details the ILS procedure.

4.2. Policy

When the process occupies state s_k, a conventional one-step lookahead policy evaluates an action x_k as the expected value of the lookahead mechanism across all possible states s_{k+1} at the next epoch. Because in our continuous-time formulation the number of such states is infinite, we instead

Figure 1 Lookahead Policy



execute the ILS procedure across a discrete set of possible future events. We consider one event for each truck en route to the warehouse. In the event, the truck arrives before all other trucks en route to the warehouse and before the next service completion. We also consider the event that the next service completion occurs before the arrival of all trucks en route to the warehouse. The cost of an action in the lookahead policy is the expected value of the ILS solutions across these events. Figure 1 depicts the decision rule. Using a decision tree, it shows the current state, the set of feasible actions, and the discrete set of possible future events that comprise the one-step lookahead. The notation used in the figure is described in the remainder of the section.

Formally, let $c(s_k^x) = \min\{c_j : j \in \mathcal{J}^p(s_k^x)\}$ be the earliest completion time of trucks in service in state s_k^x and let $j(s_k^x) = \arg\min\{c_j : j \in \mathcal{J}^p(s_k^x)\}$ be the truck that achieves the minimum. If $\mathcal{J}^p(s_k^x)$ is empty, then take $c(s_k^x)$ to be ∞. Denote by $\{s_k^x, j\}$ the event that the next arrival or service completion is triggered by truck j given that action x_k is selected in state s_k. If truck j belongs to $\mathcal{J}^d(s_k^x)$, then $\{s_k^x, j\} = \{A_j < A_{j'} \,\forall\, j' \in \mathcal{J}^d(s_k^x) \setminus \{j\} \text{ and } A_j < c(s_k^x) | s_k^x\}$. If truck $j = j(s_k^x)$, then $\{s_k^x, j\} = \{A_{j'} > c(s_k^x) \,\forall\, j' \in \mathcal{J}^d(s_k^x) | s_k^x\}$. Then, $\mathbb{E}[A_j | \{s_k^x, j\}]$ denotes the expected arrival time of truck $j \in \mathcal{J}^d(s_k^x)$ given event $\{s_k^x, j\}$, where expectation is taken with respect to the belief distributions in state s_k^x. In our experiments, we use simulation to estimate this expected value. If truck j belongs to $\mathcal{J}^d(s_k^x)$, then let $s(\{s_k^x, j\})$ be a state with time component $t = \mathbb{E}[A_j | \{s_k^x, j\}]$ and all other components as in state s_k^x. Construct a vector of arrival times with $\hat{a}_j = \mathbb{E}[A_j | \{s_k^x, j\}]$, with $\hat{a}_{j'} = \mathbb{E}[A_{j'} | A_{j'} > \hat{a}_j]$ for all $j' \in \mathcal{J}^d(s_k^x) \setminus \{j\}$, and with $\hat{a}_{j'} = a_{j'}$ as in state s_k^x for all remaining trucks $j' \in \mathcal{J} \setminus \mathcal{J}^d(s_k^x)$. If $j = j(s_k^x)$, then let $s(\{s_k^x, j\})$ be a state with time component $t = c(s_k^x)$ and all other components as in state s_k^x. Construct a vector of arrival times with $\hat{a}_{j'} = \mathbb{E}[A_{j'} | A_{j'} > c(s_k^x)]$ for all $j' \in \mathcal{J}^d(s_k^x)$ and $\hat{a}_{j'} = a_{j'}$ as in state s_k^x for all remaining trucks $j' \in \mathcal{J} \setminus \mathcal{J}^d(s_k^x)$.

Denoting the one-step lookahead policy by π_{one}, the decision rule in state s_k is

$$X_k^{\pi_{\text{one}}}(s_k) = \underset{x_k \in \mathcal{X}(s_k)}{\arg\min} \left\{ W_k(s_k, x_k) + \sum_{j \in \mathcal{J}^d(s_k^x) \cup \{j(s_k^x)\}} \hat{W}\left(s\left(\{s_k^x, j\}\right), \hat{a}\right) \mathbb{P}\left(\{s_k^x, j\}\right) \right\}. \tag{2}$$

For a given action x_k, the decision rule executes the ILS for each state $s(\{s_k^x, j\})$ and corresponding arrival times $\hat{a}$ such that j is a truck en route to the warehouse or j is the next truck to complete service. Each ILS objective value is weighted by the probability $\mathbb{P}(\{s_k^x, j\})$ of event $\{s_k^x, j\}$, then added to the contribution $W_k(s_k, x_k)$. In our experiments, these probabilities are estimated via simulation. The decision rule returns an action that achieves the minimum value.

4.3. Computational Considerations

As problem size increases, the computation required to execute the lookahead decision rules grows. To ease this burden, we use heuristic rules to reduce the number of actions, to limit the number of future states, and to decrease the complexity of the ILS procedure. As we show in the Appendix, collectively, these rules dramatically reduce computation without significant degradation of policy quality.

We reduce the number of actions in two ways. First, when one or more docks are available and one or more trucks are waiting at the warehouse, we disallow queuing if a truck can be assigned and its service can be completed before the next expected arrival. This restriction is motivated by Kanet and Sridharan (2000), who show that queuing is suboptimal in this circumstance when arrival times are deterministic. The same result does not necessarily hold in the stochastic and partially-observed setting of this paper, but our preliminary experiments indicate that such a limitation reduces computation with only a nominal effect on policy quality. Denote the queuing action by $\mathbf{0}$. It sets x_{kj} to 0 for all $j \in \mathcal{J}^w(s_k)$. We also refer to this action as the *zero* action. Let $p(s_k) = \min\{p_j : j \in \mathcal{J}^w(s_k)\}$ be the smallest service time across all trucks waiting at the warehouse for assignment in state s_k. If for each truck $j \in \mathcal{J}^d(s_k)$ en route to the warehouse its expected arrival time $\mathbb{E}[A_j]$ is greater than $t_k + p(s_k)$, where expectation is taken with respect to the belief distribution in state s_k, then the lookahead policy minimizes across $\mathcal{X}(s_k) \setminus \{\mathbf{0}\}$ instead of $\mathcal{X}(s_k)$ in Equation (2).

Second, when one or more docks are available and more than two trucks are waiting at the warehouse, we restrict the set of feasible actions to queuing, assignment of the truck with the shortest service time, and assignment of the truck with the second shortest service time. Our preliminary experiments indicate these are the most frequently selected actions. Ignoring other actions reduces computation with negligible impact on solution quality. Denote by $j_{(i)}$ the truck in $\mathcal{J}^w(s_k)$ with the i-th shortest service time $p_{j_{(i)}}$. Let $\tilde{\mathcal{X}}(s_k) = \{\mathbf{0}\} \cup \{x_k \in \mathcal{X}(s_k) : x_{kj_{(1)}} = 1 \text{ or } x_{kj_{(2)}} = 1\}$ be the subset of feasible actions that make no assignment, that assign truck $j_{(1)}$, or that assign truck $j_{(2)}$. Then, the lookahead policy decision rules minimizes across $\tilde{\mathcal{X}}(s_k)$ instead of $\mathcal{X}(s_k)$ in Equation (2). If the conditions to eliminate the queuing action are also satisfied, then the decision rules minimize across $\tilde{\mathcal{X}}(s_k) \setminus \{\mathbf{0}\}$.

We limit the number of possible future events considered in the lookahead decision rule to states whose probabilities satisfy a threshold. This eliminates ILS executions from states that make insignificant contributions to the expected value of the ILS solution, thereby reducing computation with only minor effects on policy quality. Denote the threshold by ϕ. In our experiments we set ϕ to 0.05. Then, in Equation (2), the summation is indexed over tucks in $\{j \in \mathcal{J}^d(s_k^x) \cup \{j(s_k^x)\} : \mathbb{P}(\{s_k^x, j\}) \geq \phi\}$ whose corresponding state likelihoods meet or exceed ϕ instead of over all trucks in $\mathcal{J}^d(s_k^x) \cup \{j(s_k^x)\}$.

We restrict the trucks on which the ILS operates to half the trucks en route to the warehouse plus trucks waiting at the warehouse. Our preliminary experiments indicate trucks that meet this criteria have a much higher impact on solution quality than trucks that do not. By focusing the ILS operators on these trucks, we reduce computation without any significant loss in solution quality. Denote by $\bar{j}_{(i)}$ the truck in $\mathcal{J}^d(s)$ with the i-th smallest expected arrival time $\mathbb{E}[A_{\bar{j}_{(i)}}]$, where expectation is taken with respect to the belief distribution in state s. Let $\bar{\mathcal{J}}^d(s) = \{\bar{j}_{(i)} \in \mathcal{J}^d(s) : i \leq \lfloor J/2 \rfloor\}$ be the set of $\lfloor J/2 \rfloor$ trucks in $\mathcal{J}^d(s)$ whose expected arrival times are earliest. Then, across all permutations in $\Gamma(s)$, the ILS procedure may only operate on trucks in $\bar{\mathcal{J}}^d(s) \cup \mathcal{J}^w(s)$.

5. Dual Bound

Policy quality is assessed via comparison to a lower bound. We use information relaxations coupled with an information penalty (Brown et al. 2010) to develop a dual bound on the cost of an optimal truck scheduling policy. Information in our POMDP includes ETAs, unobserved truck arrival time distributions, and actual truck arrival times. The dual bound works with this information via three filtrations: *natural*, *distribution*, and *perfect information*. Each filtration contains the same information, but presents it at different times to the decision maker. Policies are selected under the perfect information filtration, penalties utilize the distribution filtration, and the bound is evaluated via the natural filtration.

Natural filtration $\mathbb{F} = (\mathcal{F}_0, \ldots, \mathcal{F}_K)$ is given by the POMDP model. Each σ-algebra $\mathcal{F}_k$ is the information known to the decision maker at the beginning of period k. As described in §3, in state s_k this information includes ETA histories $e_j(t_k)$ plus realized arrival times a_j for each truck $j \in \mathcal{J} \setminus \mathcal{J}^d(s_k)$ already arrived to the warehouse. Distribution filtration $\mathbb{G} = (\mathcal{G}_0, \ldots, \mathcal{G}_K)$ reveals truck arrival time distributions to the decision maker at the beginning of the time horizon. Thus, under each σ-algebra $\mathcal{G}_k$, the unobserved distribution F_{A_j} is known. All other information is presented as in the natural filtration. Perfect information filtration $\mathbb{I} = (\mathcal{I}_0, \ldots, \mathcal{I}_K)$ reveals all uncertainties to the decision maker at the beginning of the time horizon. In particular, in each σ-algebra $\mathcal{I}_k$, arrival time a_j for each truck $j \in \mathcal{J}$ is known. The distribution filtration relaxes the natural filtration and the perfect information filtration relaxes the distribution filtration: at each epoch k, $\mathcal{F}_k \subseteq \mathcal{G}_k \subseteq \mathcal{I}_k$. Thus, the decision maker knows more in each period under filtration $\mathbb{G}$ than under filtration $\mathbb{F}$ and more under filtration $\mathbb{I}$ than under filtrations $\mathbb{G}$ and $\mathbb{F}$.

For clarity, in this section we augment the notation for contributions, policy sets, and policy costs to specify the associated filtration. Denote by $W_k^{\mathbb{F}}(s_k, x_k) = \mathbb{E}[\sum_{j\in\mathcal{J}} W_{kj}(s_k, x_k, t_{k+1}) | s_k, x_k, \mathcal{F}_k]$ the period-k contribution under the natural filtration, by $\Pi_{\mathbb{F}}$ the set of policies adapted to the natural filtration, and by $W_{\mathbb{F}}^{\pi} = \mathbb{E}[\sum_{k=0}^{K} W_k^{\mathbb{F}}(s_k, X_k^{\pi}(s_k)) | s_0, \mathbb{F}]$ the cost of a policy $\pi \in \Pi_{\mathbb{F}}$ under the natural filtration. In §3, these terms are introduced as $W_k(s_k, x_k)$, Π, and W^{π}, respectively. Analogously define the terms $W_k^{\mathbb{G}}(s_k, x_k)$, $\Pi_{\mathbb{G}}$, and $W_{\mathbb{G}}^{\pi}$ for the distribution filtration and $W_k^{\mathbb{I}}(s_k, x_k)$, $\Pi_{\mathbb{I}}$, and $W_{\mathbb{I}}^{\pi}$ for the perfect information filtration. Notice that with perfect information, because all uncertainties are revealed, $W_k^{\mathbb{I}}(s_k, x_k) = \sum_{j\in\mathcal{J}} W_{kj}(s_k, x_k, t_{k+1})$ and $W_{\mathbb{I}}^{\pi} = \sum_{k=0}^{K} \sum_{j\in\mathcal{J}} W_{kj}(s_k, X_k^{\pi}(s_k), t_{k+1})$ are deterministic quantities. Additionally, notice that ETAs may trigger decision epochs under all filtrations.

The remainder of this section proceeds as follows. In §5.1, we analyze the distribution filtration and identify properties that ease computation of the dual bound. In §5.2, we treat the perfect information filtration and describe an information penalty. In §5.3, we propose a compact MILP to compute the dual bound.

5.1. ETAs and Distributional Information

The analysis in this section plays a crucial intermediate step in deriving a dual bound. Because the distribution filtration reveals unobserved truck arrival time distributions, estimating them via ETAs and belief distributions is unnecessary. We leverage this to simplify optimization under the distribution filtration. First, we show that policies that assign trucks to docks at epochs triggered by

ETAs are suboptimal. Then, we demonstrate that epochs triggered by ETAs may be ignored. Finally, we formulate an MDP and show that solving it is equivalent to optimizing under the distribution filtration.

Denote by $\bar{\Pi}_{\mathbb{G}} \subseteq \Pi_{\mathbb{G}}$ the subset of policies adapted to distribution filtration $\mathbb{G}$ whose decision rules do not assign vehicles to docks at epochs triggered by ETAs. If epoch k is triggered by an ETA, then each policy $\bar{\pi} \in \bar{\Pi}_{\mathbb{G}}$ employs decision rule $X_k^{\bar{\pi}}(s_k) = \mathbf{0}$ and makes no assignment in associated state s_k. Conversely, for each policy $\pi \in \Pi_{\mathbb{G}} \setminus \bar{\Pi}_{\mathbb{G}}$, there is at least one epoch k triggered by an ETA such that decision rule $X_k^{\pi}(s_k) \neq \mathbf{0}$ assigns a truck to a dock in associated state s_k. Each policy $\bar{\pi} \in \bar{\Pi}_{\mathbb{G}}$ corresponds to a policy $\pi \in \Pi_{\mathbb{G}} \setminus \bar{\Pi}_{\mathbb{G}}$ as follows. Suppose policy π assigns truck j' to a dock in state $s_{k'}$ such that epoch k' is triggered by an ETA. Then, there must exist an earlier epoch $\bar{k}'$ such that j' belongs to the set of waiting trucks $\mathcal{J}^w(s_{\bar{k}'})$ and the zero action $X_{\bar{k}'}^{\pi}(s_{\bar{k}'}) = \mathbf{0}$ is selected, meaning truck j' was not assigned to a dock in state $s_{\bar{k}'}$, but could have been. In policy $\bar{\pi}$, the assignment of all such trucks j' to a dock is made in a corresponding state $s_{\bar{k}'}$ instead of in state $s_{k'}$.

Lemma 1 asserts that the cost $W_{\mathbb{G}}^{\bar{\pi}}$ of policy $\bar{\pi} \in \bar{\Pi}_{\mathbb{G}}$ is no larger than the cost $W_{\mathbb{G}}^{\pi}$ of a corresponding policy $\pi \in \Pi_{\mathbb{G}} \setminus \bar{\Pi}_{\mathbb{G}}$. The result follows from shifting truck assignments made in epochs triggered by ETAs to earlier epochs. Starting service sooner reduces the expected waiting time. Thus, policy π is weakly improved by a corresponding policy $\bar{\pi}$.

Lemma 1 **(Assignment at ETA Epochs)**. *For policy $\bar{\pi} \in \bar{\Pi}_{\mathbb{G}}$ and a corresponding policy $\pi \in \Pi_{\mathbb{G}} \setminus \bar{\Pi}_{\mathbb{G}}$, $W_{\mathbb{G}}^{\pi} \geq W_{\mathbb{G}}^{\bar{\pi}}$.*

Proof. Let $\boldsymbol{s}^{\pi} = (s_0^{\pi}, \ldots, s_K^{\pi})$ be a sample path induced by policy π, where $s_0^{\pi} = s_0$ is the initial state. Denote by $\mathcal{K}'(\boldsymbol{s}^{\pi})$ the set of all epochs along $\boldsymbol{s}^{\pi}$ triggered by an ETA such that policy π assigns a truck to a dock. Denote by j' the truck assigned at epoch $k' \in \mathcal{K}'(\boldsymbol{s}^{\pi})$ and the action by $x_{k'}(j')$. It sets $x_{k'j'}$ to 1 and $x_{k'j}$ to 0 for all $j \in \mathcal{J}^w(s_{k'})$ not equal to j'. For each $k' \in \mathcal{K}'(\boldsymbol{s}^{\pi})$, there must be an earlier non-ETA epoch $\bar{k}'$ such that truck j' is at the warehouse and policy π does not assign a truck to a dock. The action at epoch $\bar{k}'$ is $\mathbf{0}$.

For sample path $\boldsymbol{s}^{\pi}$, the corresponding sample path induced by policy $\bar{\pi}$ is $\boldsymbol{s}^{\bar{\pi}} = (s_0^{\bar{\pi}}, \ldots, s_K^{\bar{\pi}})$. It proceeds like sample path $\boldsymbol{s}^{\pi}$, but with the following changes. For each $k' \in \mathcal{K}'(\boldsymbol{s}^{\pi})$, move the assignment of truck j' from epoch k' to epoch $\bar{k}'$. The action at epoch $\bar{k}'$ is $x_{\bar{k}'}(j')$. It assigns truck j' to a dock in epoch $\bar{k}'$. The action at epoch k' is $\mathbf{0}$. This shifts service start time $u_{j'}$ from $t_{k'}$ to $t_{\bar{k}'}$. The change is reflected in state $s_{\bar{k}'+1}^{\bar{\pi}}$ and in each subsequent state in sample path $\boldsymbol{s}^{\bar{\pi}}$.

Changing the service start time for truck j' eliminates positive contributions due to truck j' in epochs $\bar{k}', \dots, k'-1$, which leads to the result. Denote by $\bar{\mathcal{K}}'(\boldsymbol{s}^\pi) = \{(\bar{k}', \dots, k'-1) : k' \in \mathcal{K}'(\boldsymbol{s}^\pi)\}$ the set of all epoch sequences from $\bar{k}'$ to $k'-1$ for each $k' \in \mathcal{K}'(\boldsymbol{s}^\pi)$. Let

$$g(k) = \sum_{(\bar{k}', \dots, k') \in \bar{\mathcal{K}}'(\boldsymbol{s}^\pi)} \mathbf{1}\left\{k \in (\bar{k}', \dots, k')\right\} \tag{3}$$

be the number of times epoch k appears in a sequence belonging to $\bar{\mathcal{K}}'(\boldsymbol{s}^\pi)$. Denote by $W^\pi_{\mathbb{G}}(\boldsymbol{s}^\pi) = \sum_{k=0}^K W^{\mathbb{G}}_k(s^\pi_k, X^\pi_k(s^\pi_k))$ the contributions accumulated by policy π along sample path $\boldsymbol{s}^\pi$. Define $W^{\bar{\pi}}_{\mathbb{G}}(\boldsymbol{s}^{\bar{\pi}})$ analogously. Then,

$$\begin{aligned}
W^\pi_{\mathbb{G}}\left(\boldsymbol{s}^\pi\right) &= \sum_{k=0}^K W^{\mathbb{G}}_k\left(s^\pi_k, X^\pi_k\left(s^\pi_k\right)\right) && (4)\\
&= \sum_{k=0}^K \mathbb{E}\left[\sum_{j\in\mathcal{J}} W_{kj}\left(s^\pi_k, X^\pi_k\left(s^\pi_k\right), t_{k+1}\right) \middle| s^\pi_k, X^\pi_k\left(s^\pi_k\right), \mathcal{G}_k\right] && (5)\\
&= \sum_{k=0}^K \mathbb{E}\left[\left|\left\{j \in \mathcal{J}^w\left(s^\pi_k\right) : \left(X^\pi_k\left(s^\pi_k\right)\right)_{kj} = 0\right\}\right| \left(t_{k+1} - t_k\right) \middle| s^\pi_k, X^\pi_k\left(s^\pi_k\right), \mathcal{G}_k\right] && (6)\\
&\geq \sum_{k=0}^K \mathbb{E}\left[\left(\left|\left\{j \in \mathcal{J}^w\left(s^\pi_k\right) : \left(X^\pi_k\left(s^\pi_k\right)\right)_{kj} = 0\right\}\right| - g(k)\right) \left(t_{k+1} - t_k\right) \middle| s^\pi_k, X^\pi_k\left(s^\pi_k\right), \mathcal{G}_k\right] && (7)\\
&= \sum_{k=0}^K \mathbb{E}\left[\left|\left\{j \in \mathcal{J}^w\left(s^{\bar{\pi}}_k\right) : \left(X^{\bar{\pi}}_k\left(s^{\bar{\pi}}_k\right)\right)_{kj} = 0\right\}\right| \left(t_{k+1} - t_k\right) \middle| s^{\bar{\pi}}_k, X^{\bar{\pi}}_k\left(s^{\bar{\pi}}_k\right), \mathcal{G}_k\right] && (8)\\
&= \sum_{k=0}^K \mathbb{E}\left[\sum_{j\in\mathcal{J}} W_{kj}\left(s^{\bar{\pi}}_k, X^{\bar{\pi}}_k\left(s^{\bar{\pi}}_k\right), t_{k+1}\right) \middle| s^{\bar{\pi}}_k, X^{\bar{\pi}}_k\left(s^{\bar{\pi}}_k\right), \mathcal{G}_k\right] && (9)\\
&= \sum_{k=0}^K W^{\mathbb{G}}_k\left(s^{\bar{\pi}}_k, X^{\bar{\pi}}_k\left(s^{\bar{\pi}}_k\right)\right) && (10)\\
&= W^{\bar{\pi}}_{\mathbb{G}}\left(\boldsymbol{s}^{\bar{\pi}}\right). && (11)
\end{aligned}$$

Equations (5) and (9) hold by definition. Equations (6) and (8) follow from Equation (1). Equation (7) follows from Equation (3) because, by construction, $0 \leq g(k) \leq |\{j \in \mathcal{J}^w(s^\pi_k) : (X^\pi_k(s^\pi_k))_{kj} = 0\}|$. In epoch k, policy $\bar{\pi}$ reduces the number of waiting trucks by $g(k)$ relative to the number of waiting trucks due to policy π. It follows that

$$\begin{aligned}
W^\pi_{\mathbb{G}} &= \mathbb{E}\left[\sum_{k=0}^K W^{\mathbb{G}}_k\left(s_k, X^\pi_k(s_k)\right) \middle| s_0, \mathbb{G}\right] && (12)\\
&= \mathbb{E}\left[W^\pi_{\mathbb{G}}\left(\boldsymbol{s}^\pi\right) \middle| \mathbb{G}\right] && (13)
\end{aligned}$$

$$\geq \mathbb{E}\left[W^{\bar{\pi}}_{\mathbb{G}}\left(\boldsymbol{s}^{\bar{\pi}}\right)\middle|\mathbb{G}\right] \tag{14}$$

$$= \mathbb{E}\left[\sum_{k=0}^{K} W^{\mathbb{G}}_{k}\left(s_k, X^{\bar{\pi}}_{k}(s_k)\right)\middle|s_0, \mathbb{G}\right] \tag{15}$$

$$= W^{\bar{\pi}}_{\mathbb{G}}. \tag{16}$$

Equations (12) and (16) hold by definition. Equations (13) and (15) express the expected sum of contributions as the expected contribution across sample paths. Equation (14) follows from two facts. First, Equations (4)–(11) demonstrate that $W^{\pi}_{\mathbb{G}}(\boldsymbol{s}^{\pi}) \geq W^{\bar{\pi}}_{\mathbb{G}}(\boldsymbol{s}^{\bar{\pi}})$ for any sample path $\boldsymbol{s}^{\pi}$ and the corresponding sample path $\boldsymbol{s}^{\bar{\pi}}$. Second, the exogenous information (ETAs and arrival times) is the same across the two paths, and thus they occur with the same probability. Consequently, $\mathbb{E}[W^{\pi}_{\mathbb{G}}(\boldsymbol{s}^{\pi})|\mathbb{G}] \geq \mathbb{E}[W^{\bar{\pi}}_{\mathbb{G}}(\boldsymbol{s}^{\bar{\pi}})|\mathbb{G}]$. □

Beyond Lemma 1, we show that the cost of a policy in $\bar{\Pi}_{\mathbb{G}}$ may be calculated without consideration of epochs triggered by ETAs. Define an MDP similar to the POMDP in §3, but without ETAs: epochs are not triggered by ETAs, the state variable does not include ETA histories nor belief distributions, the time of the next epoch is not a function of future ETAs, ETAs do not play a role in contributions, and information unfolds according to distribution filtration $\mathbb{G}$. Epochs are triggered by truck arrivals, service completions, and trucks at the yard when docks are available. The state $\tilde{s}_k = (\tilde{t}_k, (a_j, u_j)_{j\in\mathcal{J}})$ at epoch k includes the time at which the epoch occurs, truck arrival times, and service start times. The set of feasible actions $\mathcal{X}(\tilde{s}_k)$ in state $\tilde{s}_k$ is the same as the set of feasible actions in the POMDP. A transition to epoch $k+1$ occurs at random time $\tilde{t}_{k+1} = \min\{\bar{A}_k, \bar{p}_k, \bar{t}_k\}$. A transition updates arrival times and service start times as in the POMDP. Contribution $W^{\mathbb{G}}_{k}(\tilde{s}_k, x_k) = \mathbb{E}[\sum_{j\in\mathcal{J}} W_{kj}(\tilde{s}_k, x_k, \tilde{t}_{k+1})|\tilde{s}_k, x_k, \mathcal{G}_k]$ calculates expected waiting time as a function of unobserved arrival time distributions. Denote the set of policies by $\tilde{\Pi}_{\mathbb{G}}$. The cost $W^{\tilde{\pi}}_{\mathbb{G}} = \mathbb{E}[\sum_{k=0}^{K} W^{\mathbb{G}}_{k}(\tilde{s}_k, X^{\tilde{\pi}}_{k}(\tilde{s}_k))|\tilde{s}_0, \mathbb{G}]$ of a policy $\tilde{\pi} \in \tilde{\Pi}_{\mathbb{G}}$ is the expected sum of contributions conditional on the distribution filtration.

For each policy $\bar{\pi} \in \bar{\Pi}_{\mathbb{G}}$, there is a corresponding policy $\tilde{\pi} \in \tilde{\Pi}_{\mathbb{G}}$ consisting of the decision rules in $\bar{\pi}$ for all states tied to epochs that are not triggered by ETAs. As we show in Lemma 2, for any policy in $\bar{\Pi}_{\mathbb{G}}$, decision rules for states tied to epochs triggered by ETAs are inconsequential. Lemma 2 asserts that the cost $\tilde{W}^{\tilde{\pi}}_{\mathbb{G}}$ of policy $\tilde{\pi}$ is equal to the cost $W^{\bar{\pi}}_{\mathbb{G}}$ of policy $\bar{\pi}$. The result follows from the zero action selected by policy $\bar{\pi}$ in each epoch triggered by an ETA. The proof shows that contributions accrued at epochs triggered by ETAs are captured by excluding ETAs from the definition of the next epoch time $\tilde{t}_{k+1}$. Thus, instead of working with the POMDP under the distribution filtration, we may instead work with the MDP.

Lemma 2 (**Cost and ETA Epochs**). *For policy $\bar{\pi} \in \bar{\Pi}_{\mathbb{G}}$ and the corresponding policy $\tilde{\pi} \in \tilde{\Pi}_{\mathbb{G}}$, $W_{\mathbb{G}}^{\bar{\pi}} = W_{\mathbb{G}}^{\tilde{\pi}}$.*

Proof. The proof is in five parts, labeled (i)–(v). We prove four intermediate results then the main result. Let s_k and s_{k+n} be two states on a sample path induced by policy $\bar{\pi}$ and separated by the maximum possible number of consecutive epochs $k+1,\ldots,k+n-1$ triggered by ETAs. Denote the associated history of states, actions, and information by $\bar{\mathbf{s}} = (s_k, X_k^{\bar{\pi}}(s_k), t_{k+1} = \bar{e}_k, \ldots, s_{k+n-2}, X_{k+n-2}^{\bar{\pi}}(s_{k+n-2}), t_{k+n-1} = \bar{e}_{k+n-2}, s_{k+n-1}, X_{k+n-1}^{\bar{\pi}}(s_{k+n-1}), t_{k+n} \neq \bar{e}_{k+n-1}, s_{k+n})$, where all epochs are triggered by ETAs except for epoch $k+n$. For $k \le l \le k+n$, denote by $\bar{\mathbf{s}}(l) = (s_k, X_k^{\bar{\pi}}(s_k), t_{k+1} = \bar{e}_k, \ldots, s_l)$ the history $\bar{\mathbf{s}}$ through state s_l. The definition of $\bar{\mathbf{s}}$ implies n is chosen such that $\mathbb{P}(t_{k+n} = \bar{e}_{k+n-1} | \bar{\mathbf{s}}(k+n-1)) = 0$ and $\mathbb{P}(t_{k+n} \neq \bar{e}_{k+n-1} | \bar{\mathbf{s}}(k+n-1)) = 1$, meaning the probability of an additional epoch triggered by an ETA occurring before an epoch triggered by a different event is zero. In other words, time t_{k+n} must be determined by an event that is not an ETA. Additionally, the definition of $\bar{\mathbf{s}}$ implies that arrival times $(a_j)_{j\in\mathcal{J}}$ are the same in states $s_k,\ldots,s_{k+n-1}$ and that service start times $(u_j)_{j\in\mathcal{J}}$ are the same in states $s_{k+1},\ldots,s_{k+n-1}$. This is true because epochs $k+1,\ldots,k+n-1$ are not triggered by truck arrivals and actions taken in the associated states make no assignments.

Along history $\bar{\mathbf{s}}$, for $l = k+1,\ldots,k+n-1$, the first three intermediate results are

(i) $\mathcal{J}^w\left(s_l^x\right) = \mathcal{J}^w\left(s_k^x\right)$

(ii) $\mathbb{E}\left[\sum_{j\in\mathcal{J}} W_{l,j}\left(s_l, X_l^{\bar{\pi}}(s_l), t_{l+1}\right) \middle| \bar{\mathbf{s}}(l), X_l^{\bar{\pi}}(s_l), \mathcal{G}_l\right] = \mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} t_{l+1} - \bar{e}_{l-1} \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_l\right]$

(iii) $\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} t_{l+1} - \bar{e}_{l-1} \middle| s_k, X_k^{\bar{\pi}}(s_k), t_{l+1} \neq \bar{e}_l, \mathcal{G}_l\right] = \mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k, \bar{p}_k, \bar{t}_k\right\} - \bar{e}_{l-1} \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_l\right]$.

Proof of Part (i). Epoch $k+1$ is triggered by an ETA at time $t_{k+1} = \bar{e}_k$. Consequently, epoch $k+1$ does not signal the arrival of a truck, and thus the set of trucks $\mathcal{J}^w(s_{k+1})$ waiting at the warehouse in state s_{k+1} is equal to the set of trucks $\mathcal{J}^w(s_k^x)$ waiting at the warehouse in post-decision state s_k^x. Further, because $X_{k+1}^{\bar{\pi}}(s_{k+1}) = \mathbf{0}$ makes no assignment, the set of waiting trucks does not change. Thus, $\mathcal{J}^w(s_{k+1}^x) = \mathcal{J}^w(s_k^x)$. Similarly, each subsequent epoch $l = k+2,\ldots,k+n-1$ is triggered by an ETA at time $t_l = \bar{e}_{l-1}$ and the zero action $X_l^{\bar{\pi}}(s_l) = \mathbf{0}$ is selected. Because these events and actions do not change the set of trucks waiting at the warehouse, $\mathcal{J}^w(s_l^x) = \mathcal{J}^w(s_k^x)$ for $l = k+1,\ldots,k+n-1$.

Proof of Part (ii). Per Equation (1), the contribution is zero for any truck $j \in \mathcal{J}^w(s_k)$ such that x_{lj} is not equal to zero. Equivalently, per the definitions of $\mathcal{J}^w(\cdot)$ and the post-decision state, the contribution of any truck not belonging to $\mathcal{J}^w(s_l^x)$ is zero. Thus, summation over the trucks in $\mathcal{J}$ is equivalent to summation over the trucks in $\mathcal{J}^w(s_l^x)$. Summation across $\mathcal{J}^w(s_l^x) = \mathcal{J}^w(s_k^x)$ follows from part (i). To understand the change in condition and value, notice that after action $X_k^{\bar{\pi}}(s_k)$ is selected in state s_k, subsequent actions and information in history $\bar{s}$ do not result in any changes to the state variable that effect expected contributions. Because the zero action makes no assignment, per the transition function, it has no effect on the state variable. Further, under distribution filtration $\mathbb{G}$, the decision maker knows unobserved truck arrival time distributions. Thus, even though ETA histories and belief distributions are different across each state in the history, they do not impact the expectation, which is a function of the unobserved distributions. Finally, the times t_k and $t_l = \bar{e}_{l-1}$ at which the process occupies states s_k and s_l, respectively, are different. However, this is accounted for in the value $t_{l+1} - \bar{e}_{l-1}$, which follows from Equation (1) and the condition on $\bar{s}(l)$. Thus, conditioning on s_k and $X_k^{\bar{\pi}}(s_k)$ plus setting $t_l = \bar{e}_{l-1}$ result in the same expected value as conditioning on $\bar{s}(l)$ and $X_l^{\bar{\pi}}(s_l) = \mathbf{0}$.

Proof of Part (iii). Per the condition $t_{l+1} \neq \bar{e}_l$, $t_{l+1} = \min\{\bar{A}_l, \bar{p}_l, \bar{t}_l\}$. Because epochs $k+1, \ldots, l$ in history $\bar{s}$ are triggered by ETAs, and because the zero actions selected in these periods make no assignments, $\bar{A}_l = \bar{A}_k$, $\bar{p}_l = \bar{p}_k$, and $\bar{t}_l = \bar{t}_k$. Thus, $t_{l+1} = \min\{\bar{A}_k, \bar{p}_k, \bar{t}_k\}$.

The fourth intermediate result is

$$\textbf{(iv)} \quad \mathbb{E}\left[\sum_{l=k}^{k+n-1} W_l^{\mathbb{G}}\left(s_l, X_l^{\bar{\pi}}(s_l)\right) \middle| s_k, \mathbb{G}\right] = \mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k, \bar{p}_k, \bar{t}_k\right\} - t_k \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_k\right].$$

Proof of Part (iv).

$$\mathbb{E}\left[\sum_{l=k}^{k+n-1} W_l^{\mathbb{G}}\left(s_l, X_l^{\bar{\pi}}(s_l)\right) \middle| s_k, \mathbb{G}\right]$$

$$= \mathbb{E}\left[\sum_{l=k}^{k+n-1} \mathbb{E}\left[\sum_{j \in \mathcal{J}} W_{lj}\left(s_l, X_l^{\bar{\pi}}(s_l), t_{l+1}\right) \middle| \bar{s}(l), X_l^{\bar{\pi}}(s_l), \mathcal{G}_l\right] \middle| s_k, \mathbb{G}\right] \tag{17}$$

$$= \mathbb{E}\left[\mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} t_{k+1} - t_k \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_k\right] + \sum_{l=k+1}^{k+n-1} \mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} t_{l+1} - \bar{e}_{l-1} \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_l\right] \middle| s_k, \mathbb{G}\right] \tag{18}$$

$$= \mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} t_{k+1} - t_k \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_k\right] + \sum_{l=k+1}^{k+n-1} \mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} t_{l+1} - \bar{e}_{l-1} \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_l\right] \prod_{g=k+1}^{l} \mathbb{P}\left(t_g = \bar{e}_{g-1} | \bar{s}(g-1)\right) \tag{19}$$

$$= \mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} t_{k+1} - t_k \middle| s_k, X_k^{\bar{\pi}}(s_k), t_{k+1} = \bar{e}_k, \mathcal{G}_k\right] \mathbb{P}\left(t_{k+1} = \bar{e}_k | s_k\right)$$

$$
\begin{aligned}
&+\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} t_{k+1}-t_k \middle| s_k, X_k^{\bar{\pi}}(s_k), t_{k+1}\neq \bar{e}_k, \mathcal{G}_k\right]\mathbb{P}\left(t_{k+1}\neq \bar{e}_k|s_k\right)\\
&+\sum_{l=k+1}^{k+n-1}\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} t_{l+1}-\bar{e}_{l-1}\middle| s_k, X_k^{\bar{\pi}}(s_k), t_{l+1}=\bar{e}_l, \mathcal{G}_l\right]\mathbb{P}\left(t_{l+1}=\bar{e}_l|\bar{s}(l)\right)\prod_{g=k+1}^{l}\mathbb{P}\left(t_g=\bar{e}_{g-1}|\bar{s}(g-1)\right)\\
&+\sum_{l=k+1}^{k+n-1}\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} t_{l+1}-\bar{e}_{l-1}\middle| s_k, X_k^{\bar{\pi}}(s_k), t_{l+1}\neq\bar{e}_l, \mathcal{G}_l\right]\mathbb{P}\left(t_{l+1}\neq\bar{e}_l|\bar{s}(l)\right)\prod_{g=k+1}^{l}\mathbb{P}\left(t_g=\bar{e}_{g-1}|\bar{s}(g-1)\right) \quad (20)\\
=&\,\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} \bar{e}_k-t_k\middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_k\right]\mathbb{P}\left(t_{k+1}=\bar{e}_k|s_k\right)\\
&+\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k,\bar{p}_k,\bar{t}_k\right\}-t_k\middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_k\right]\mathbb{P}\left(t_{k+1}\neq\bar{e}_k|s_k\right)\\
&+\sum_{l=k+1}^{k+n-2}\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} \bar{e}_l-\bar{e}_{l-1}\middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_l\right]\mathbb{P}\left(t_{l+1}=\bar{e}_l|\bar{s}(l)\right)\prod_{g=k+1}^{l}\mathbb{P}\left(t_g=\bar{e}_{g-1}|\bar{s}(g-1)\right)\\
&+\sum_{l=k+1}^{k+n-2}\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k,\bar{p}_k,\bar{t}_k\right\}-\bar{e}_{l-1}\middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_l\right]\mathbb{P}\left(t_{l+1}\neq\bar{e}_l|\bar{s}(l)\right)\prod_{g=k+1}^{l}\mathbb{P}\left(t_g=\bar{e}_{g-1}|\bar{s}(g-1)\right)\\
&+\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k,\bar{p}_k,\bar{t}_k\right\}-\bar{e}_{k+n-2}\middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_{k+n-1}\right]\\
&\quad\cdot\mathbb{P}\left(t_{k+n-1}=\bar{e}_{k+n-2}|\bar{s}(k+n-2)\right)\prod_{g=k+1}^{k+n-2}\mathbb{P}\left(t_g=\bar{e}_{g-1}|\bar{s}(g-1)\right) \quad (21)\\
=&\,\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k,\bar{p}_k,\bar{t}_k\right\}-t_k\middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_k\right]. \quad (22)
\end{aligned}
$$

Equation (17) holds by definition. Per the transition function, conditioning on history $\bar{s}(l)$ in the inner expectation is equivalent to conditioning on state s_l. Equation (18) follows from part (ii). Equation (19) calculates the outer expectation conditional on distribution filtration $\mathbb{G}$ and on each epoch subsequent to k being triggered by an ETA, or equivalently, the history $\bar{s}(l)$. Equation (20) holds by the law of total expectation. Equation (21) holds by part (iii), the conditions on t_{l+1}, the assumption that $\mathbb{P}(t_{k+n}=\bar{e}_{k+n-1}|\bar{s}(k+n-1))=0$, and the assumption that $\mathbb{P}(t_{k+n}\neq\bar{e}_{k+n-1}|\bar{s}(k+n-1))=1$. Equation (22) is obtained by manipulating Equation (21) as follows. Combine the terms in one period with the terms in the previous period. Begin with periods $k+n-1$ and $k+n-2$:

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k,\bar{p}_k,\bar{t}_k\right\}-\bar{e}_{k+n-2}\middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_{k+n-1}\right]\\
&\quad\cdot\mathbb{P}\left(t_{k+n-1}=\bar{e}_{k+n-2}|\bar{s}(k+n-2)\right)\prod_{g=k+1}^{k+n-2}\mathbb{P}\left(t_g=\bar{e}_{g-1}|\bar{s}(g-1)\right) \quad (23)\\
&+\mathbb{E}\left[\sum_{j\in\mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k,\bar{p}_k,\bar{t}_k\right\}-\bar{e}_{k+n-3}\middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_{k+n-2}\right]
\end{aligned}
$$

$$\cdot \mathbb{P}\left(t_{k+n-1} \neq \bar{e}_{k+n-2} | \bar{s}(k+n-2)\right) \prod_{g=k+1}^{k+n-2} \mathbb{P}\left(t_g = \bar{e}_{g-1} | \bar{s}(g-1)\right) \tag{24}$$

$$+ \mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} \bar{e}_{k+n-2} - \bar{e}_{k+n-3} \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_{k+n-2} \right]$$

$$\cdot \mathbb{P}\left(t_{k+n-1} = \bar{e}_{k+n-2} | \bar{s}(k+n-2)\right) \prod_{g=k+1}^{k+n-2} \mathbb{P}\left(t_g = \bar{e}_{g-1} | \bar{s}(g-1)\right) \tag{25}$$

$$= \mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k, \bar{p}_k, \bar{t}_k\right\} - \bar{e}_{k+n-3} \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_{k+n-2} \right] \prod_{g=k+1}^{k+n-2} \mathbb{P}\left(t_g = \bar{e}_{g-1} | \bar{s}(g-1)\right). \tag{26}$$

The expectations in Equations (23) and (25) are the same whether conditioned on $\mathcal{G}_{k+n-2}$ or $\mathcal{G}_{k+n-1}$. Thus, the terms may be combined to eliminate $\bar{e}_{k+n-2}$ and the resulting expectation may be conditioned on $\mathcal{G}_{k+n-2}$. Factoring the common terms with Equation (24) leaves $\mathbb{P}(t_{k+n-1} = \bar{e}_{k+n-2}|\bar{s}(k+n-2)) + \mathbb{P}(t_{k+n-1} \neq \bar{e}_{k+n-2}|\bar{s}(k+n-2)) = 1$, which yields Equation (26). Proceeding backwards in the this fashion eventually reduces Equation (21) to

$$\mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k, \bar{p}_k, \bar{t}_k\right\} - \bar{e}_k \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_{k+1} \right] \mathbb{P}\left(t_{k+1} = \bar{e}_k | s_k\right) \tag{27}$$

$$+ \mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} \min\left\{\bar{A}_k, \bar{p}_k, \bar{t}_k\right\} - t_k \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_k \right] \mathbb{P}\left(t_{k+1} \neq \bar{e}_k | s_k\right) \tag{28}$$

$$+ \mathbb{E}\left[\sum_{j \in \mathcal{J}^w(s_k^x)} \bar{e}_k - t_k \middle| s_k, X_k^{\bar{\pi}}(s_k), \mathcal{G}_k \right] \mathbb{P}\left(t_{k+1} = \bar{e}_k | s_k\right). \tag{29}$$

Combining the terms in Equations (27)-(29) in the same way as the terms in Equations (23)-(25) yields Equation (22).

Part (v). Part (iv) establishes that the expected sum of contributions up until the first epoch after k not triggered by an ETA is equal to the expected time elapsed across waiting trucks until the first epoch after k not triggered by an ETA. Thus, if the POMDP model under distribution filtration $\mathbb{G}$ is modified to exclude epochs triggered by ETAs and to calculate contributions via $\tilde{t}_{k+1} = \min\{\bar{A}_k, \bar{p}_k, \bar{t}_k\}$ instead of $t_{k+1} = \min\{\bar{A}_k, \bar{e}_k, \bar{p}_k, \bar{t}_k\}$, then the cost of policy $\bar{\pi}$ is unchanged. Because the MDP model triggers epochs and calculates contributions in exactly this way, the cost of the corresponding policy $\tilde{\pi}$ is the same: $W_{\mathbb{G}}^{\bar{\pi}} = W_{\mathbb{G}}^{\tilde{\pi}}$. □

The primary result of this section shows that optimization via the POMDP under the distribution filtration is equivalent to optimization via the MDP. Denote by $W_{\mathbb{G}}^{\star} = \min\{W_{\mathbb{G}}^{\pi} : \pi \in \Pi_{\mathbb{G}}\}$ the cost of an optimal POMDP policy under filtration $\mathbb{G}$ and by $\tilde{W}_{\mathbb{G}}^{\star} = \min\{W_{\mathbb{G}}^{\tilde{\pi}} : \tilde{\pi} \in \tilde{\Pi}_{\mathbb{G}}\}$ the cost of an optimal MDP policy. Proposition 1 asserts that $W_{\mathbb{G}}^{\star}$ equals $\tilde{W}_{\mathbb{G}}^{\star}$. The result follows directly from Lemmas 1 and 2. Proposition 1 is a crucial intermediate step to obtaining a dual bound. Equivalence of the MDP and the POMDP under the distribution filtration allows us to substitute one model for the other. Switching models permits the exclusion of epochs triggered by ETAs. Because such epochs are action-dependent, disregarding them eases computation.

Proposition 1 **(MDP).** $W^{\star}_{\mathbb{G}} = \tilde{W}^{\star}_{\mathbb{G}}$.

Proof.

$$W^{\star}_{\mathbb{G}} = \min\left\{W^{\pi}_{\mathbb{G}} : \pi \in \Pi_{\mathbb{G}}\right\} \tag{30}$$

$$= \min\left\{W^{\pi}_{\mathbb{G}} : \pi \in \left\{\Pi_{\mathbb{G}} \setminus \bar{\Pi}_{\mathbb{G}}\right\} \cup \bar{\Pi}_{\mathbb{G}}\right\} \tag{31}$$

$$= \min\left\{W^{\bar{\pi}}_{\mathbb{G}} : \bar{\pi} \in \bar{\Pi}_{\mathbb{G}}\right\} \tag{32}$$

$$= \min\left\{W^{\tilde{\pi}}_{\mathbb{G}} : \tilde{\pi} \in \tilde{\Pi}_{\mathbb{G}}\right\} \tag{33}$$

$$= \tilde{W}^{\star}_{\mathbb{G}}. \tag{34}$$

Equations (30) and (34) follow from definitions. Equation (31) partitions the policy space. Equation (32) follows from Lemma 1. For every policy $\pi \in \Pi_{\mathbb{G}} \setminus \bar{\Pi}_{\mathbb{G}}$, there is a corresponding policy $\bar{\pi} \in \bar{\Pi}_{\mathbb{G}}$ such that $W^{\pi}_{\mathbb{G}} \geq W^{\bar{\pi}}_{\mathbb{G}}$. Thus, a minimizer must exist in $\bar{\Pi}_{\mathbb{G}}$. Equation (33) follows from Lemma 2. For every $\bar{\pi} \in \bar{\Pi}_{\mathbb{G}}$, the corresponding $\tilde{\pi} \in \tilde{\Pi}_{\mathbb{G}}$ has the same cost. Thus, minimizing across policies in $\bar{\Pi}_{\mathbb{G}}$ yields a policy with the same cost as minimizing across $\tilde{\Pi}_{\mathbb{G}}$. □

5.2. Deriving the Bound

The dual bound allows the decision maker to select a policy under perfect information filtration $\mathbb{I}$ in response to any trajectory in natural filtration $\mathbb{F}$, but requires that contributions be accrued per distribution filtration $\mathbb{G}$. The bound penalizes the perfect-information contribution function $W^{\mathbb{I}}_k(\tilde{s}_k, x_k)$ in each period. We follow the template for information penalties proposed by Brown et al. (2010). Their work points to penalties that approximate the value of using knowledge about the future. Such penalties discourage clairvoyant decisions in the present. Our approximation is the contribution function evaluated under the perfect information filtration less the contribution function evaluated under the distribution filtration: $W^{\mathbb{I}}_k(\tilde{s}_k, x_k) - W^{\mathbb{G}}_k(\tilde{s}_k, x_k)$. Subtracting the penalty and canceling terms yields $W^{\mathbb{I}}_k(\tilde{s}_k, x_k) - (W^{\mathbb{I}}_k(\tilde{s}_k, x_k) - W^{\mathbb{G}}_k(\tilde{s}_k, x_k)) = W^{\mathbb{G}}_k(\tilde{s}_k, x_k)$. The result is the contribution function evaluated under filtration $\mathbb{G}$ instead of filtration $\mathbb{I}$. Even though the bound permits action selection with full foresight, the penalized period-k contribution incurs waiting time as if the immediate future is uncertain. This penalty is similar to the smoothing penalty used in Brown and Smith (2022).

Notice that $W_k^{\mathbb{I}}(s_k, x_k) - W_k^{\mathbb{F}}(s_k, x_k)$ is also a valid penalty. Using the natural filtration and the POMDP state variable satisfies the requirements of Brown et al. (2010) and works with the analysis below. However, penalizing via the distribution filtration allows us to work with MDP contributions instead of with POMDP contributions, and therefore to ignore decision epochs triggered by ETAs. Because the occurrence and timing of such epochs are policy-dependent, removing them from consideration markedly simplifies the optimization. This advantage facilitates the MILP presented in the subsequent section.

Denote a realization of truck arrival times in the natural filtration by $a = (a_j)_{j \in \mathcal{J}}$. Denote by $\tilde{\Pi}_{\mathbb{I}}$ the MDP policy set adapted to the perfect information filtration. Applying the penalty across all epochs and optimizing across $\tilde{\Pi}_{\mathbb{I}}$ leads to penalized problem

$$f(a) = \min_{\tilde{\pi} \in \tilde{\Pi}_{\mathbb{I}}} \left\{ \sum_{k=0}^{K} W_k^{\mathbb{G}} \left(\tilde{s}_k, X_k^{\tilde{\pi}}(\tilde{s}_k) \right) \right\}. \tag{35}$$

The quantity $f(a)$ is the minimum cost of a policy in $\tilde{\Pi}_{\mathbb{I}}$ chosen with full knowledge of truck arrival times a, but with waiting times accrued as if the time of the next epoch is governed by distribution filtration $\mathbb{G}$. In other words, the penalty negates the benefit of knowing how future truck arrival times determine the time of the next epoch, which in turn negates the benefit of knowing the waiting time in the current period.

Denote by $W_{\mathrm{P}}^{\star} = \mathbb{E}[f(a)|\mathbb{F}]$ the expected value of penalized problem $f(a)$ under the natural filtration. Proposition 2 asserts that $W_{\mathrm{P}}^{\star}$ is a lower bound on the cost $W^{\star}$ of an optimal policy for the POMDP. The result follows from Proposition 1 and the theory of information relaxations and duality (Brown et al. 2010). The proof of Proposition 2 shows that $W^{\star} \geq \mathbb{E}[\tilde{W}_{\mathbb{G}}^{\star}|\mathbb{F}] \geq W_{\mathrm{P}}^{\star}$. Thus, the expected value of an optimal MDP policy sits between the cost of an optimal POMDP policy and the expected value of the penalized problem. Consequently, any gap between $W_{\mathrm{P}}^{\star}$ and the cost of a candidate POMDP policy is due to the suboptimality of the policy, the additional cost of partial observability relative to the MDP, a loose bound, or some combination of these factors. Our computational experiments indicate, however, that $W_{\mathrm{P}}^{\star}$ is often tight, in the sense that it is approximately equal to $\mathbb{E}[\tilde{W}_{\mathbb{G}}^{\star}|\mathbb{F}]$.

PROPOSITION 2 **(Dual Bound).** $W^{\star} \geq W_P^{\star}$.

Proof.

$$
\begin{aligned}
W^\star &= \min_{\pi\in\Pi_{\mathbb{F}}} \mathbb{E}\left[\sum_{k=0}^{K} W_k^{\mathbb{F}}\left(s_k, X_k^\pi(s_k)\right) \middle| s_0, \mathbb{F}\right] && (36)\\
&\geq \mathbb{E}\left[\min_{\pi\in\Pi_{\mathbb{G}}} \mathbb{E}\left[\sum_{k=0}^{K} W_k^{\mathbb{G}}\left(s_k, X_k^\pi(s_k)\right) \middle| s_0, \mathbb{G}\right] \middle| \mathbb{F}\right] && (37)\\
&= \mathbb{E}\left[W_{\mathbb{G}}^\star \middle| \mathbb{F}\right] && (38)\\
&= \mathbb{E}\left[\tilde{W}_{\mathbb{G}}^\star \middle| \mathbb{F}\right] && (39)\\
&= \mathbb{E}\left[\min_{\tilde{\pi}\in\tilde{\Pi}_{\mathbb{G}}} \mathbb{E}\left[\sum_{k=0}^{K} W_k^{\mathbb{G}}\left(\tilde{s}_k, X_k^{\tilde{\pi}}(\tilde{s}_k)\right) \middle| s_0, \mathbb{G}\right] \middle| \mathbb{F}\right] && (40)\\
&\geq \mathbb{E}\left[\min_{\tilde{\pi}\in\tilde{\Pi}_{\mathbb{I}}} \left\{\sum_{k=0}^{K} W_k^{\mathbb{I}}\left(\tilde{s}_k, X_k^{\tilde{\pi}}(\tilde{s}_k)\right) - \left(W_k^{\mathbb{I}}\left(\tilde{s}_k, X_k^{\tilde{\pi}}(\tilde{s}_k)\right) - W_k^{\mathbb{G}}\left(\tilde{s}_k, X_k^{\tilde{\pi}}(\tilde{s}_k)\right)\right)\right\} \middle| \mathbb{F}\right] && (41)\\
&= \mathbb{E}\left[\min_{\tilde{\pi}\in\tilde{\Pi}_{\mathbb{I}}} \left\{\sum_{k=0}^{K} W_k^{\mathbb{G}}\left(\tilde{s}_k, X_k^{\tilde{\pi}}(\tilde{s}_k)\right)\right\} \middle| \mathbb{F}\right] && (42)\\
&= \mathbb{E}\left[f(a) \middle| \mathbb{F}\right] && (43)\\
&= W_{\mathrm{P}}^\star. && (44)
\end{aligned}
$$

Equations (36), (38), (40), (43), and (44) hold by definition. Equation (37) follows from Lemma 2.1 of Brown et al. (2010) because distribution filtration $\mathbb{G}$ relaxes natural filtration $\mathbb{F}$ and the zero penalty is dual feasible. Equation (39) follows from Proposition 1 because for every trajectory of states in the natural filtration, $W_{\mathbb{G}}^\star = \tilde{W}_{\mathbb{G}}^\star$. The inequality in Equation (41) holds by Lemma 2.1 and Proposition 2.2 of Brown et al. (2010): perfect information filtration $\mathbb{I}$ relaxes distribution filtration $\mathbb{G}$ and the penalty is dual feasible when the contribution function is taken to be the generating function. Equation (42) cancels terms. □

5.3. Solving the Penalized Problem

This section focuses on solving the penalized problem. In contrast to the POMDP, which models dynamic decisions in the face of unobserved uncertainties, the penalized problem is all but deterministic. Although each period's contribution is accrued in expectation, because all information is revealed at the beginning of the time horizon, decisions across epochs can be made concurrently instead of in sequence. Thus, the penalized problem is primarily a combinatorial optimization task and can be approached through the lens of mathematical programming. We begin this section with an alternative dynamic program formulation. Then, using simulation to estimate the contribution, we formulate a sample average approximation of the alternative dynamic program. Finally, we model the sample average approximation as a MILP.

The purpose of the alternative dynamic program formulation of the penalized problem is to state the optimization in a form that is more amenable to representation as a MILP. The alternative formulation consolidates assignments made in consecutive epochs at the same time to a single epoch. In other words, it replaces sequences of single-truck assignments to docks by one assignment of multiple trucks to multiple docks. To accomplish this, the alternative formulation removes epochs triggered by the assignment of a truck to a dock when at least one truck is in the yard and at least one dock is available. In §3, such epochs are labeled as event (iv). For brevity, we refer to these epochs as *back-to-back* epochs. The alternative formulation also relaxes the action space to permit assignment of trucks up to the number of available docks.

Formally, epochs are triggered by truck arrivals and service completions. There are $2J$ epochs $0, \dots, K = 2J - 1$, one for each truck's arrival and another for its service completion. The state $\hat{s}_k = (\hat{t}_k, (a_j, u_j)_{j \in \mathcal{J}})$ at epoch k includes the time at which the epoch occurs, known truck arrival times, and service start times. The set of feasible actions $\hat{\mathcal{X}}(\hat{s}_k) = \{x_k \in \{0,1\}^{|\mathcal{J}^w(\hat{s}_k)|} : \sum_{j \in \mathcal{J}^w(\hat{s}_k)} x_{kj} \leq \bar{D}_k(\hat{s}_k)\}$ in state $\hat{s}_k$ consists of all possible assignments of trucks at the yard to available docks. Epoch $k+1$ occurs at time $\hat{t}_{k+1} = \min\{\bar{A}_k, \bar{p}_k\}$, the smaller of the next truck arrival time and the next service completion time. As in the original formulation of the penalized problem, the time of the next epoch is random with respect to contributions and known for the purpose of decision making. Contribution $W_k^{\mathbb{G}}(\hat{s}_k, x_k) = \mathbb{E}[\sum_{j \in \mathcal{J}} W_{kj}(\hat{s}_k, x_k, \hat{t}_{k+1}) | \hat{s}_k, x_k, \mathcal{G}_k]$ is the expected waiting time conditional on the distribution filtration. The transition sets service start time $u_j = \hat{t}_k$ for each truck $j \in \mathcal{J}^w(\hat{s}_k)$ such that $x_{kj} = 1$. Denote the set of policies by $\hat{\Pi}_{\mathbb{I}}$. Then, the alternative formulation of the penalized problem is

$$h(a) = \min_{\hat{\pi} \in \hat{\Pi}_{\mathbb{I}}} \left\{ \sum_{k=0}^{K} W_k^{\mathbb{G}} \left(\hat{s}_k, X_k^{\hat{\pi}}(\hat{s}_k) \right) \right\}. \tag{45}$$

For each policy $\tilde{\pi} \in \tilde{\Pi}_{\mathbb{I}}$, there is a corresponding policy $\hat{\pi} \in \hat{\Pi}_{\mathbb{I}}$. States tied to back-to-back epochs do not exist in the alternative formulation. The actions taken by policy $\tilde{\pi}$ in these states are consolidated into a single action taken by policy $\hat{\pi}$. Let $\tilde{s}_{k+1}, \dots, \tilde{s}_{k+n}$ be a sequence of n such states induced by policy $\tilde{\pi}$. Let $\tilde{s}_k$ be the preceding state and denote by $\hat{s}_k = \tilde{s}_k$ the same state induced by policy $\hat{\pi}$. The action $X_k^{\hat{\pi}}(\hat{s}_k) = \sum_{l=k}^{k+n} X_l^{\tilde{\pi}}(\tilde{s}_l)$ taken by policy $\hat{\pi}$ in state $\hat{s}_k$ is the component-wise sum of the actions taken by policy $\tilde{\pi}$ across states $\tilde{s}_k, \dots, \tilde{s}_{k+n}$. In all other states, the decision rules in policy $\hat{\pi}$ are the same as the decision rules in policy $\tilde{\pi}$.

Proposition 3 asserts that the cost $h(a)$ of an optimal policy in the alternative formulation is equal to the cost $f(a)$ of an optimal policy in the original formulation. The result follows from the construction of the consolidated action and the fact that zero expected time elapses between back-to-back epochs. Proposition 3 facilitates the transition to a MILP by reducing the number of epoch triggers in exchange for a larger action space. Although the reformulated action space is combinatorial, this is naturally captured in the MILP.

PROPOSITION 3 **(Back-to-Back Epochs).** $h(a) = f(a)$.

Proof. Consider a policy $\tilde{\pi} \in \tilde{\Pi}_{\mathbb{I}}$ and the corresponding policy $\hat{\pi} \in \hat{\Pi}_{\mathbb{I}}$. Let $\tilde{s}_{k+1}, \ldots, \tilde{s}_{k+n}$ be a sequence of states induced by policy $\tilde{\pi}$ such that the associated epochs are back-to-back epochs. Let $\tilde{s}_k$ be the preceding state and denote by $\hat{s}_k = \tilde{s}_k$ the same state induced by policy $\hat{\pi}$. The proof is in three parts, labeled (i)–(iii).

Part (i). For $l = k, \ldots, k+n-1$,

$$W_l^{\mathbb{G}}\left(\tilde{s}_l, X_l^{\tilde{\pi}}(\tilde{s}_l)\right) = \mathbb{E}\left[\sum_{j\in\mathcal{J}} W_{lj}\left(\tilde{s}_l, X_l^{\tilde{\pi}}(\tilde{s}_l), \tilde{t}_{l+1}\right) \middle| \tilde{s}_l, X_l^{\tilde{\pi}}(\tilde{s}_l), \mathcal{G}_l\right] \tag{46}$$

$$= \mathbb{E}\left[\sum_{j\in\mathcal{J}} (\tilde{t}_{l+1} - \tilde{t}_l)\, \mathbf{1}\left\{j \in \mathcal{J}^w(\tilde{s}_l) \text{ and } \left(X_l^{\tilde{\pi}}(\tilde{s}_l)\right)_{lj} = 0\right\} \middle| \tilde{s}_l, X_l^{\tilde{\pi}}(\tilde{s}_l), \mathcal{G}_l\right] \tag{47}$$

$$= 0. \tag{48}$$

Equations (46) and (47) hold by definition of the contribution. Because states $\tilde{s}_{k+1}, \ldots, \tilde{s}_k$ are triggered by back-to-back epochs, then by the transition function, $\tilde{t}_m = \tilde{t}_k$ for $m = k+1, \ldots, k+n$. Consequently, the difference $\tilde{t}_{l+1} - \tilde{t}_l$ in Equation (47) is zero. This establishes Equation (48).

Part (ii).

$$\sum_{l=k}^{k+n} W_l^{\mathbb{G}}\left(\tilde{s}_l, X_l^{\tilde{\pi}}(\tilde{s}_l)\right) = W_{k+n}^{\mathbb{G}}\left(\tilde{s}_{k+n}, X_{k+n}^{\tilde{\pi}}(\tilde{s}_{k+n})\right) \tag{49}$$

$$= W_k^{\mathbb{G}}\left(\hat{s}_k, X_k^{\hat{\pi}}(\hat{s}_k)\right). \tag{50}$$

Equation (49) holds by part (i). Equation (50) follows from two facts. First, post-decision states $\tilde{s}_{k+n}^x$ and $\hat{s}_k^x$ are the same. To see this, check each component of the state variable. As shown in part (i), $\tilde{t}_{k+n} = \tilde{t}_k$. Further, by assumption, $\hat{t}_k = \tilde{t}_k$. Thus, $\tilde{t}_{k+n} = \hat{t}_k$. Because the transition to the post-decision state does not modify the time component of the state variable, these quantities remain equal in the post-decision state. Because truck arrival times are known from the beginning of the time horizon,

a is the same in both states. Finally, via the construction of action $X_k^{\hat{\pi}}(\hat{s}_k) = \sum_{l=k}^{k+n} X_l^{\tilde{\pi}}(\tilde{s}_l)$, both formulations assign the same trucks to start service at the same time. Thus, the vector of service start times $(u_j)_{j \in \mathcal{J}}$ in each post-decision state is the same. Second, the time $\tilde{t}_{k+n+1}$ of the next epoch in the original formulation is the same as the time $\hat{t}_{k+1}$ of the next epoch in the alternative formulation. To see this, notice that $\bar{A}_k = \bar{A}_{k+n}$. This is true because, by assumption, epochs $k+1, \ldots, k+n$ are not triggered by truck arrivals. Further, because $\tilde{s}_{k+n}^x = \hat{s}_k^x$, it follows that $\bar{p}_{k+n}$ in the original formulation is the same as $\bar{p}_k$ in the alternative formulation. Finally, by assumption, $\bar{t}_{k+n}$ is greater than or equal to $\bar{A}_{k+n}$ and $\bar{p}_{k+n}$. Thus, $\tilde{t}_{k+n+1} = \min\{\bar{A}_{k+n}, \bar{p}_{k+n}, \bar{t}_{k+n}\} = \min\{\bar{A}_{k+n}, \bar{p}_{k+n}\}$, which is equal to $\hat{t}_{k+1} = \min\{\bar{A}_k, \bar{p}_k\}$. These two facts establish Equation (50).

Part (iii). Notice that action $X_k^{\hat{\pi}}(\hat{s}_k)$ belongs to $\hat{\mathcal{X}}(\hat{s}_k)$ by construction. The number of trucks assigned to docks in the original formulation is $\sum_{l=k}^{k+n} \sum_{j \in \mathcal{J}^w(\tilde{s}_l)} (X_l^{\tilde{\pi}}(\tilde{s}_l))_{lj}$. Because each action is feasible in the original formulation, this number of trucks must be less than or equal to the sum of available docks $\sum_{l=k}^{k+n} \bar{D}_l(\tilde{s}_l)$, which, per the transition function and by the assumption of back-to-back epochs is the same as $\bar{D}_k(\hat{s}_k)$. Thus, $X_k^{\hat{\pi}}(\hat{s}_k)$ is feasible in the alternative formulation. Then, it follows from part (ii) that any sequence of states $\tilde{s}_k, \ldots, \tilde{s}_{k+n}$ in the original formulation, along with the associated actions $X_k^{\tilde{\pi}}(\tilde{s}_k), \ldots, X_{k+n}^{\tilde{\pi}}(\tilde{s}_{k+n})$, incur the same cost as the corresponding state $\hat{s}_k$ in the alternative formulation, plus the associated action $X_k^{\hat{\pi}}(\hat{s}_k)$. Because the two policies take the same actions in all other states, it must be that the cost of policy $\tilde{\pi}$ is the same as the cost of policy $\hat{\pi}$: $\sum_{k=0}^{K} W_k^{\mathbb{G}}(\tilde{s}_k, X_k^{\tilde{\pi}}(\tilde{s}_k)) = \sum_{k=0}^{K} W_k^{\mathbb{G}}(\hat{s}_k, X_k^{\hat{\pi}}(\hat{s}_k))$. Finally, by construction there is a one-to-one correspondence between policies in $\tilde{\Pi}_{\mathbb{I}}$ and $\hat{\Pi}_{\mathbb{I}}$, and thus the cost of an optimal policy in each model must be the same:

$$f(a) = \min_{\tilde{\pi} \in \tilde{\Pi}_{\mathbb{I}}} \left\{ \sum_{k=0}^{K} W_k^{\mathbb{G}} \left(\tilde{s}_k, X_k^{\tilde{\pi}}(\tilde{s}_k) \right) \right\} = \min_{\hat{\pi} \in \hat{\Pi}_{\mathbb{I}}} \left\{ \sum_{k=0}^{K} W_k^{\mathbb{G}} \left(\hat{s}_k, X_k^{\hat{\pi}}(\hat{s}_k) \right) \right\} = h(a). \tag{51}$$

□

We use sample average approximation (Kleywegt et al. 2002) to estimate the cost $h(a)$ of the alternative dynamic program. We estimate the period-k contribution $W_k^{\mathbb{G}}(\hat{s}_k, x_k)$ under distribution filtration $\mathbb{G}$ via simulation. For each truck $j \in \mathcal{J}$, draw N arrival times from unobserved distribution F_{A_j}. Organize the simulated arrival times into N trajectories. Trajectory $\hat{a}^n = (\hat{a}_1^n, \ldots, \hat{a}_J^n)$ is a sequence of simulated arrival times, one for each truck. Connect a simulated arrival time trajectory to decision epoch times as follows. Denote by $\bar{A}_k^n = \min\{\hat{a}_j^n : j \in \mathcal{J}^d(\hat{s}_k^x)\}$ the smallest simulated arrival time in trajectory n across trucks en route to the warehouse in state $\hat{s}_k^x$. When the process occupies state $\hat{s}_k$ and action x_k is selected, denote by $\hat{t}^n_{k+1} = \min\{\bar{A}^n_k, \bar{p}_k\}$ the time of the next decision epoch when truck arrival times follow trajectory $\hat{a}^n$. Let

$$W^n_{kj}(\hat{s}_k, x_k, \hat{t}^n_{k+1}) = \begin{cases} \hat{t}^n_{k+1} - \hat{t}_k, \ j \in \mathcal{J}^w(\hat{s}_k) \text{ and } x_{kj} = 0, \\ 0, \text{ otherwise,} \end{cases} \tag{52}$$

be the waiting time incurred by truck j when the process occupies state $\hat{s}_k$, action x_k is selected, and the next epoch begins at time $\hat{t}^n_{k+1}$. Then, $\hat{W}^{\mathbb{G}}_k(\hat{s}_k, x_k) = 1/N \sum_{n=1}^N \sum_{j \in \mathcal{J}} W^n_{kj}(\hat{s}_k, x_k, \hat{t}^n_{k+1})$ is an unbiased and consistent estimate of $W^{\mathbb{G}}_k(\hat{s}_k, x_k)$. Thus, we approximate the penalized problem as

$$\hat{h}(a) = \min_{\hat{\pi} \in \hat{\Pi}_{\mathbb{I}}} \left\{ \sum_{k=0}^{K} \hat{W}^{\mathbb{G}}_k \left(\hat{s}_k, X^{\hat{\pi}}_k(\hat{s}_k) \right) \right\}. \tag{53}$$

As the number of samples N grows, $\hat{h}(a)$ converges to $h(a)$ (Kleywegt et al. 2002).

Equations (54)–(113) model as a MILP the sample average approximation in Equation (53). Table 1 states the purpose of each variable in the MILP and the discussion below outlines the objective and constraints. Denote by $b(a)$ the cost of an optimal MILP solution. Proposition 4 asserts that if arrival times a are unique, then $b(a)$ equals the cost $\hat{h}(a)$ of the sample average approximation. The result follows from using algebraic variables and constraints to express the epochs, states, actions, transitions, and contributions in the sample average approximation. The requirement of unique truck arrival times simplifies the MILP and is not a limitation. When we use simulation to estimate dual bound $W^\star_{\mathrm{P}}$, it is easy to check for duplicates. However, because we model arrival times as continuous random variables, the probability of drawing two or more identical arrival times is zero.

$$\textbf{Minimize:} \quad \sum_{k=0}^{K-1} \left[\frac{1}{N} \sum_{n=1}^{N} \left(\sum_{j=1}^{J} \delta^n_{jk} \right) \right] \tag{54}$$

Subject To:

Scheduling

$$\sum_{d \in \mathcal{D}} x_{jd} = 1 \qquad \forall j \in \mathcal{J} \tag{55}$$

$$x_{id} + x_{jd} - y_{ij} - y_{ji} \le 1 \qquad \forall (i,j) \in \mathcal{J}, i \neq j, \forall d \in \mathcal{D} \tag{56}$$

$$x_{id} + x_{jh} + y_{ij} + y_{ji} \le 2 \qquad \forall (i,j) \in \mathcal{J}, i \neq j, \forall (d,h) \in \mathcal{D}, d \neq h \tag{57}$$

Table 1 MIP Variables and Definitions

Variable	Definition
$\mathcal{K}$	set of decision epochs $\{0,\ldots,K=2J-1\}$
u_j	start time of truck j
x_{jd}	1 if truck j is assigned to dock d, 0 otherwise
$y_{jj'}$	1 if truck j precedes truck j' at the same dock, 0 otherwise
c_j	completion time of truck j
$z_{jj'}$	1 if start time of truck j equals completion time of truck j', 0 otherwise
$\lambda_{jj'}$	1 if start time of truck j equals arrival time of truck j', 0 otherwise
t_k	time at decision epoch k
q_{jk}	1 if decision epoch k is the arrival of truck j, 0 otherwise
r_{jk}	1 if decision epoch k is the completion of truck j, 0 otherwise
l_k	1 if decision epoch k is triggered by a completion, 0 otherwise
τ_k	time of the next service completion after decision epoch k
v^n_{jk}	1 if sample $\hat{a}^n_j > t_k$ and $a_j > t_k$, 0 otherwise
α^n_{jk}	1 if $\hat{a}^n_j \geq t_k - \epsilon$, 0 otherwise
β_{jk}	1 if $a_j \geq t_k - \epsilon$, 0 otherwise
w_{jk}	1 if truck j is waiting at decision epoch k, 0 otherwise
h_{jk}	1 if t_k is greater or equal than a_j, 0 otherwise
g_{jk}	1 if t_k is less than u_j, 0 otherwise
m^n_k	minimum of τ_k and sampled arrivals in the set $\{\hat{a}^n_j, j \in \mathcal{J}, n \in \mathcal{N} : v^n_{jk} = 1\}$
η^n_{jk}	1 if $m^n_k = \hat{a}^n_j$, 0 otherwise
γ^n_k	1 if $m^n_k = \tau_k$, 0 otherwise
δ^n_{jk}	waiting time for sample n at decision epoch k for truck j

$$u_i + p_i x_{id} - M(1 - y_{ij}) \leq u_j \qquad \forall (i,j) \in \mathcal{J}, i \neq j, \forall d \in \mathcal{D} \tag{58}$$

$$a_j \leq u_j \qquad \forall j \in \mathcal{J} \tag{59}$$

Service Start Time

$$c_j = p_j + u_j \qquad \forall j \in \mathcal{J} \tag{60}$$

$$u_j \leq c_{j'} + M(1 - z_{jj'}) \qquad \forall (j,j') \in \mathcal{J}, j' \neq j \tag{61}$$

$$u_j \geq c_{j'} - M(1 - z_{jj'}) \qquad \forall (j,j') \in \mathcal{J}, j' \neq j \tag{62}$$

$$u_j \leq a_{j'} + M(1 - \lambda_{jj'}) \qquad \forall (j,j') \in \mathcal{J}, j' \neq j \tag{63}$$

$$u_j \geq a_{j'} - M(1 - \lambda_{jj'}) \qquad \forall (j,j') \in \mathcal{J}, j' \neq j \tag{64}$$

$$\sum_{j' \in \mathcal{J}, j' \neq j} z_{jj'} + \sum_{j' \in \mathcal{J}} \lambda_{jj'} = 1 \qquad \forall j \in \mathcal{J} \tag{65}$$

Decision Epochs

$$t_{k+1} \geq t_k \qquad \forall k \in \mathcal{K} \setminus \{K\} \tag{66}$$

$$t_k \geq a_j - M(1 - q_{jk}) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{67}$$

$$t_k \le a_j + M(1 - q_{jk}) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{68}$$

$$t_k \ge c_j - M(1 - r_{jk}) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{69}$$

$$t_k \le c_j + M(1 - r_{jk}) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{70}$$

$$\sum_{k \in \mathcal{K}} q_{jk} = 1 \qquad \forall j \in \mathcal{J} \tag{71}$$

$$\sum_{k \in \mathcal{K}} r_{jk} = 1 \qquad \forall j \in \mathcal{J} \tag{72}$$

$$\sum_{j \in \mathcal{J}} q_{jk} + \sum_{j \in \mathcal{J}} r_{jk} = 1 \qquad \forall k \in \mathcal{K} \tag{73}$$

Next Service Completion

$$l_k = \sum_{j \in \mathcal{J}} r_{jk} \qquad \forall k \in \mathcal{K} \tag{74}$$

$$\tau_k \ge t_{k+1} - M(1 - l_{k+1}) \qquad \forall k \in \mathcal{K} \setminus K \tag{75}$$

$$\tau_k \ge t_{k+i} - M\left(1 - l_{k+i} + \sum_{j=1}^{i-1} l_{k+j}\right) \qquad \forall k \in \mathcal{K} \setminus K, i = \{2, \ldots, K-k\} \tag{76}$$

$$\tau_k \le t_{k+1} + M(1 - l_{k+1}) \qquad \forall k \in \mathcal{K} \setminus K \tag{77}$$

$$\tau_k \le t_{k+i} + M\left(1 - l_{k+i} + \sum_{j=1}^{i-1} l_{k+j}\right) \qquad \forall k \in \mathcal{K} \setminus K, i = \{2, \ldots, K-k\} \tag{78}$$

Trucks Waiting

$$t_k \ge a_j - M(1 - h_{jk}) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{79}$$

$$t_k \le a_j - \epsilon + M h_{jk} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{80}$$

$$t_k \le u_j - \epsilon + M(1 - g_{jk}) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{81}$$

$$t_k \ge u_j - M g_{jk} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{82}$$

$$w_{jk} \le h_{jk} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{83}$$

$$w_{jk} \le g_{jk} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{84}$$

$$w_{jk} \ge h_{jk} + g_{jk} - 1 \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{85}$$

Arrival Time Samples

$$\hat{a}_j^n - t_k \ge \epsilon - M(1 - \alpha_{jk}^n) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{86}$$

$$\hat{a}_j^n - t_k \le M \alpha_{jk}^n \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{87}$$

$$a_j - t_k \geq \epsilon - M(1-\beta_{jk}) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{88}$$

$$a_j - t_k \leq M\beta_{jk} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{89}$$

$$v_{jk}^n \leq \alpha_{jk}^n \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{90}$$

$$v_{jk}^n \leq \beta_{jk} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{91}$$

$$v_{jk}^n \geq \alpha_{jk}^n + \beta_{jk} - 1 \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{92}$$

Waiting Times

$$m_k^n \leq \tau_k \qquad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{93}$$

$$m_k^n \geq \tau_k - M(1-\gamma_k^n) \qquad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{94}$$

$$m_k^n \leq \hat{a}_j^n + M(1-v_{jk}^n) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{95}$$

$$m_k^n \geq \hat{a}_j^n - M(1-\eta_{jk}^n) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{96}$$

$$\sum_{j\in\mathcal{J}} \eta_{jk}^n + \gamma_k^n = 1 \qquad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{97}$$

$$\eta_{jk}^n \leq v_{jk}^n \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{98}$$

$$\delta_{jk}^n \leq M w_{jk} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{99}$$

$$\delta_{jk}^n \leq m_k^n - t_k \qquad \forall k \in \mathcal{K}, \forall n \in \mathcal{N}, \forall j \in \mathcal{J} \tag{100}$$

$$\delta_{jk}^n \geq m_k^n - t_k - M(1-w_{jk}) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{101}$$

Decision Variables

$$u_j \geq 0 \qquad \forall j \in \mathcal{J} \tag{102}$$

$$t_k \geq 0 \qquad \forall k \in \mathcal{K} \tag{103}$$

$$\tau_k \geq 0 \qquad \forall k \in \mathcal{K} \setminus K \tag{104}$$

$$m_k^n \geq 0 \qquad \forall n \in \mathcal{N}, \forall k \in \mathcal{K} \tag{105}$$

$$\delta_{jk}^n \geq 0 \qquad \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall j \in \mathcal{J} \tag{106}$$

$$l_k \in \{0,1\} \qquad \forall k \in \mathcal{K} \tag{107}$$

$$x_{jd} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall d \in \mathcal{D} \tag{108}$$

$$y_{jj'}, \lambda_{jj'}, \in \{0,1\} \qquad \forall (j,j') \in \mathcal{J}, j \neq j' \tag{109}$$

$$z_{jj'} \in \{0,1\} \qquad \forall (j,j') \in \mathcal{J}, j \neq j' \tag{110}$$

$$\gamma_k^n \in \{0,1\} \qquad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{111}$$

$$q_{jk}, r_{jk}, \beta_{jk}, h_{jk}, g_{jk}, w_{jk} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{112}$$

$$\eta^n_{jk}, \alpha^n_{jk}, v^n_{jk} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{113}$$

Scheduling. Constraints (55) require that each truck receive service at exactly one dock. Constraints (56) ensure that if both trucks i and j are assigned to dock d, then one truck must precede the other. Constraints (57) require that precedence variables y_{ij} and y_{ji} be zero if trucks i and j are assigned to different docks. At each dock, Constraints (58) ensure that the end of service for one truck is not larger than the start of service for the subsequent truck, where $M = \max_{j\in\mathcal{J}}\{a_j\} + \sum_{j\in\mathcal{J}} p_j$ is a large number set to the largest arrival time plus the sum of service times. Constraints (59) ensure that a truck's service start time occurs at or after its arrival time.

Service Start Time. Constraints (60) define a truck's completion time as its service start time plus service time. Constraints (61) and (62) ensure that the service start time of truck j equals the service completion time of truck j' if truck j begins service immediately after truck j' finishes service. Similarly, Constraints (63) and (64) ensure that the service start time of truck j equals the arrival time of truck j' if truck j begins service immediately after j' arrives. Constraints (65) require a truck's service start times to coincide with an arrival time or a service completion time.

Decision Epochs. Constraints (66) order decision epochs by ensuring that the time of a given epoch is at least as large as the time of the previous epoch. Constraints (67) and (68) ensure that the time of decision epoch k equals the arrival time of truck j if epoch k is triggered by the arrival of truck j. Similarly, Constraints (69) and (70) ensure that the time of decision epoch k equals the service completion time of truck j if the epoch is triggered by the service completion of truck j. Constraints (71) and (72) require that each truck triggers exactly one decision epoch tied to its arrival and another decision epoch tied to its service completion. Constraints (73) require that each decision epoch is triggered by a truck arrival or a service completion, but not by both.

Next Service Completion. Constraints (74) define l_k as an indicator for whether decision epoch k is triggered by a service completion. Constraints (75) and (76) ensure that τ_k is at least the time of the next service completion after decision epoch k, and Constraints (77) and (78) ensure that τ_k is at most this value. Constraints (75) and (77) treat epoch $k+1$ while Constraints (76) and (78) address subsequent epochs.

Trucks Waiting. Constraints (79) and (80) identify truck arrival times less than or equal to the time of each decision epoch. Constraints (81) and (82) identify service start times greater than the time of each decision epoch. To implement the strict inequalities, we set ϵ to 0.00001.

Constraints (83), (84), and (85) identify trucks that satisfy both conditions: the decision epoch time is greater than or equal to the arrival time, but less than the service start time.

Arrival Time Samples. Constraints (86) and (87) identify arrival time samples greater than the time of each decision epoch. Constraints (88) and (89) identify arrival time realizations greater than the time of each decision epoch. Constraints (90), (91), and (92) identify arrival time samples such that both conditions are satisfied: the arrival time sample and the arrival time realization are both greater than the time of each decision epoch.

Waiting Times. For each arrival time sample, Constraints (93)–(98) work together to calculate the smaller of the next service completion time and the arrival time samples identified by Constraints (86)–(92). Variable m_k^n represents the minimum for sample n at epoch k. Then, Constraints (99)–(101) calculate the waiting time for truck j in sample n at epoch k as δ_{jk}^n.

Objective. Equation (54) calculates the objective as the sum of δ_{jk}^n across all epochs, all samples, and all trucks.

PROPOSITION 4 **(Penalized Problem)**. *If truck arrival times a are unique, then $b(a) = \hat{h}(a)$.*

Proof.

$$\hat{h}(a) = \min_{\hat{\pi}\in\hat{\Pi}_{\mathbb{I}}} \left\{ \sum_{k=0}^{K} \hat{W}_k^{\mathbb{G}} \left(\hat{s}_k, X_k^{\hat{\pi}}(\hat{s}_k) \right) \right\} \tag{114}$$

$$= \min_{\hat{\pi}\in\hat{\Pi}_{\mathbb{I}}} \left\{ \sum_{k=0}^{K} \frac{1}{N} \sum_{n=1}^{N} \sum_{j\in\mathcal{J}} W_{kj}^n \left(\hat{s}_k, X_k^{\hat{\pi}}(\hat{s}_k), \hat{t}_{k+1}^n \right) \right\} \tag{115}$$

$$= \min_{\hat{\pi}\in\hat{\Pi}_{\mathbb{I}}} \left\{ \sum_{k=0}^{K} \frac{1}{N} \sum_{n=1}^{N} \sum_{j\in\mathcal{J}} (\hat{t}_{k+1}^n - \hat{t}_k)\, \mathbf{1} \left\{ j \in \mathcal{J}^w(\hat{s}_k) \text{ and } \left(X_k^{\hat{\pi}}(\hat{s}_k) \right)_{kj} = 0 \right\} \right\} \tag{116}$$

$$= \min_{\hat{\pi}\in\hat{\Pi}_{\mathbb{I}}} \left\{ \sum_{k=0}^{K} \frac{1}{N} \sum_{n=1}^{N} \sum_{j\in\mathcal{J}} \left(\min\left\{ \bar{A}_k^n, \bar{p}_k \right\} - \hat{t}_k \right) \mathbf{1} \left\{ a_j \le \hat{t}_k < u_j^{\hat{\pi}} \right\} \right\} \tag{117}$$

$$= \min \left\{ \sum_{k=0}^{K} \frac{1}{N} \sum_{n=1}^{N} \sum_{j\in\mathcal{J}} \left(\min\left\{ \left\{ \hat{a}_j^n : a_j > t_k, j \in \mathcal{J} \right\} \cup \tau_k \right\} - t_k \right) w_{jk} : \right.$$

$$\left. \vphantom{\sum_{k=0}^{K}} (55)\text{–}(92), (102)\text{–}(104), (107)\text{–}(110), (112)\text{–}(113) \right\} \tag{118}$$

$$= \min \left\{ \sum_{k=0}^{K-1} \frac{1}{N} \sum_{n=1}^{N} \sum_{j\in\mathcal{J}} \delta_{jk}^n : (55)\text{–}(113) \right\} \tag{119}$$

$$= b(a) \tag{120}$$

Equation (114) is the definition of $\hat{h}(a)$. Equation (115) follows from the definition of $\hat{W}_k^{\mathbb{G}}(\hat{s}_k, X_k^{\hat{\pi}}(\hat{s}_k))$. Equation (116) follows from the definition of $W_{kj}^n(\hat{s}_k, X_k^{\hat{\pi}}(\hat{s}_k), \hat{t}_{k+1}^n)$. The difference between epoch times in Equation (117) follows from the definition of $\hat{t}_{k+1}^n$. The change in the indicator function expresses the condition that truck j is waiting and not assigned as the condition that arrival is at or before $\hat{t}_k$ and assignment is after $\hat{t}_k$. Variable $u_j^{\hat{\pi}}$ is the service start time of truck j in policy $\hat{\pi}$. Per the definitions of $\mathcal{J}^w(\cdot)$ and the zero action, the conditions are equivalent. Equation (118) transitions from optimization over policies to a mathematical program expressed as an objective subject to constraints. The term $\min\{\{\hat{a}_j^n : a_j > t_k, j \in \mathcal{J}\} \cup \tau_k\}$ is the smallest of sampled arrival times for trucks en route and the next service completion. This captures the smaller of $\bar{A}_k^n$ and $\bar{p}_k$. Variable w_{jk} captures the indicator function and the listed constraints model decision epochs, states, actions, transitions, and the estimated contribution. The assumption that truck arrival times are unique ensures that epochs tied to arrivals do not occur at the same time, which in turn ensures that contributions at arrival epochs are correctly calculated. Equation (119) captures the objective via variables δ_{jk}^n. Epoch K is not included in the outer summation because the contribution is zero. Epoch K corresponds to the final service completion. At time t_K, because there are no trucks in the yard, the expected waiting time is zero. Equation (120) follows from the definition of the MILP. □

6. Computational Experiments

This section details our computational experience with the lookahead policy and and dual bound. We demonstrate the strength of both and their real-world utility. Problem instances are outlined in §6.1, benchmarks are described in §6.2, and results are presented in §6.3.

6.1. Problem Instances

At the time of this writing, we did not have adequate nondisclosure agreements with Poste Italiene's carriers, who own the ETA data. Although we could not directly use this data, the instances we construct are representative of real scenarios encountered at Poste warehouses.

A problem instance specifies the number of trucks J, the number of docks D, a service time p_j for each truck $j \in \mathcal{J}$, an unobserved arrival time distribution F_{A_j} for each truck $j \in \mathcal{J}$, and a process governing ETAs for each truck $j \in \mathcal{J}$. We consider values of D ranging from 1 to 10 and values of J ranging from 10 to 50. Scenarios across this range are typical in practice. Some facilities consist of only one dock and many facilities consist of just a few. As mentioned in §1, warehouses with many docks typically partition the docks into operational groups. In the case of Poste Italiane, a group may consist of as many as 10 docks.

For a given number of trucks J, each service time p_j is chosen randomly from a triangular distribution with support $[10, 100]$ and mode 20. This skews service times toward the lower end of the support, but leaves open the possibility that some trucks require much longer to complete service. Each truck arrival time A_j follows a truncated normal distribution with underlying mean $\hat{\mu}_j$, underlying variance $\hat{b}_j = 20$, and support $[0, \infty)$. Arrivals are connected to service times and the number of docks by randomly choosing $\hat{\mu}_j$ from the range $[0, \theta \sum_{j \in \mathcal{J}} p_j / D]$. The parameter θ controls traffic intensity. Small values concentrate truck arrivals at the beginning of the time horizon. In this scenario, which approaches deterministic machine scheduling problems, assignment of trucks to docks in the order of shortest service time works well (Pinedo 2022). Large values spread out truck arrivals to the point that waiting for assignment is unnecessary. We set θ to 0.5. This value results in enough traffic to require non-trivial decision making. ETAs for truck j are drawn from a normal distribution with mean $\mu_j^e = \hat{\mu}_j + \kappa$ and standard deviation $\sigma_j^e = 1$. Noise parameter κ is normally distributed with mean zero and variance η. For the bulk of our experiments, we set η to 1. ETAs occur at a fixed frequency of once every time unit for each truck.

Belief distributions are updated via constrained Kalman filtering (Simon and Simon 2010). The method applies the conventional Kalman filter (Särkkä 2013, ch. 4.3) and truncates predictions to satisfy the requirement that the arrival time of a vehicle en route be greater than the current time. The Kalman filter assumes arrival time and ETAs for each truck j are normally distributed random variables with known measurement noise. The noise is normally distributed with mean zero and variance $r_j = 100$. Updating the belief distribution proceeds as follows. Let t and $t' > t$ denote the times of two consecutive ETAs for truck j and let $e_{t'}$ be the ETA at time t'. The filter for truck j takes the probability $\mathbb{P}(A_j | e_j(t'))$ of arrival time conditional on ETAs through time t' to be normal with mean $\mu_{t'j} = \mu_{tj} + G_{tj}(e_{t'} - \mu_{tj})$ and variance $b_{t'j} = (1 - G_{tj}) b_{tj}$, where $G_{tj} = b_{tj} / (b_{tj} + r_j)$ is the Kalman gain. Belief distribution $F_{A_j}(t')$ is obtained by truncating $\mathbb{P}(A_j | e_j(t'))$ at current time t'. We take initial belief distribution $F_{A_j}(t_0)$ for truck j to be truncated normal with underlying mean equal to the most recent ETA for truck j, with underlying standard deviation equal to 20 plus a random noise drawn from the standard normal distribution, and with support $[0, \infty)$.

To ensure that the belief distribution converges to the unobserved distribution, we run the Kalman filter backwards across the final updates. Let T_j be the time of the last ETA before arrival time a_j. Set $\mu_{T_j j} = \hat{\mu}_j$ and $b_{T_j j} = \hat{b}_j$. This initializes the backwards application of the filter. Then, given

$\mu_{t'j}$, $b_{t'j}$, and $e_{t'}$, the mean and variance at time t are obtained from the forward equations as $\mu_{tj} = \mu_{t'j} - b_{tj}(e_{t'} - \mu_{t'j})/r_j$ and $b_{tj} = r_j b_{t'j}/(r_j - b_{t'j})$, respectively. Begin by setting t' to T_j, then follow this trajectory for the final 20 updates. Earlier updates follow the Kalman filter equations in the forward direction.

6.2. Benchmarks

We use benchmarks to assess the quality of the lookahead policy and dual bound. We compare the lookahead policy to the dual bound and to the cost of a first-come-first-served (fcfs) benchmark, a policy often used in practice for dynamic truck scheduling. Policy π_{fcfs} selects actions based on arrival times. At each decision epoch, the truck at the yard with the earliest arrival is assigned to a dock. To measure the benefit of dual bound $W^\star_{\text{P}}$, we compare it to the *expected value with perfect information* (EVPI). In the context of §5, the EVPI is the expected value of optimal policies under perfect information filtration $\mathbb{I}$ with zero penalty: $\mathbb{E}[\min_{\hat\pi\in\hat\Pi_{\mathbb{I}}}\{\sum_{k=0}^{K} W_k^{\mathbb{I}}(\hat s_k, X_k^{\hat\pi}(\hat s_k))\}|\mathbb{F}]$. The inner optimization is a scheduling problem with release dates and identical machines. We formulate this problem as a MILP that minimizes waiting time across all trucks subject to scheduling constraints: $\min\{\sum_{j\in\mathcal{J}} u_j - a_j : (55)\text{–}(59), (102), (108), (109)\}$. It is straightforward to extend the analyses in §5 to show that $W^\star \geq$ EVPI. A positive gap between $W^\star_{\text{P}}$ and the EVPI highlights the advantage of the information penalty.

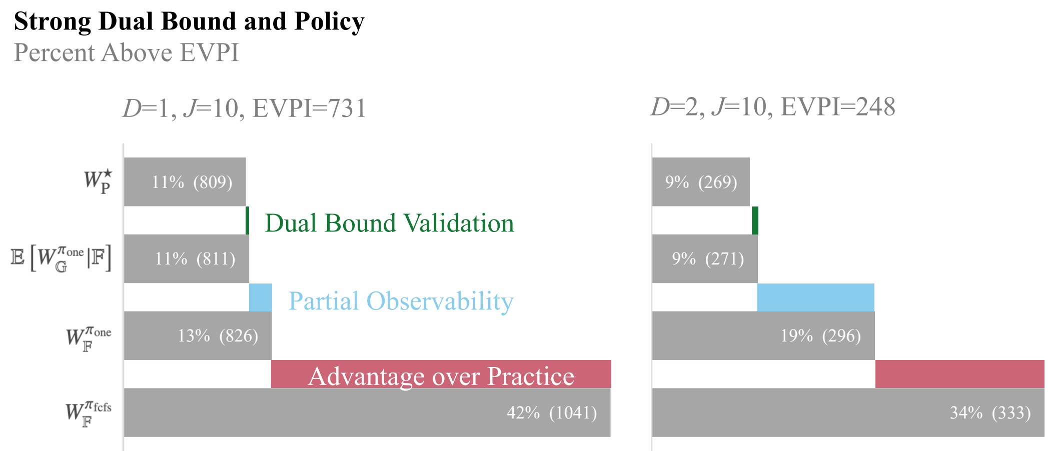
6.3. Results and Discussion

We conduct four groups of experiments. The first group examines the quality of the lookahead policy and dual bound. The second explores policy performance at scale. The third group assess the utility of filtering. The fourth group looks at the affect of ETA accuracy.

In the Appendix, we justify use of the lookahead mechanism that drives the one-step lookahead policy. We compare the ILS procedure underlying the mechanism to exact solutions and examine the computational acceleration strategies for decision rule execution described in §4.3. We conclude that the ILS procedure returns high quality solutions and that the acceleration strategies significantly decrease the computation required to execute decision rules with only nominal loss in quality.

Policy costs and dual bounds are estimated via simulation. For a given problem instance, we sample 10 information trajectories, then execute the policy or dual bound across each. Average performance over the trajectories provides an unbiased estimate. The sample average approximation used to estimate the cost of the penalized problem employs $N = 15$ trajectories. Throughout our

Figure 2 Dual Bounds and Policy Costs



experiments, for a given number of docks D and trucks J, values are an average across 15 instances with randomly generated service times, arrival time distributions, and ETAs. Thus, a single value represents the average across 10 information trajectories for each of 15 instances.

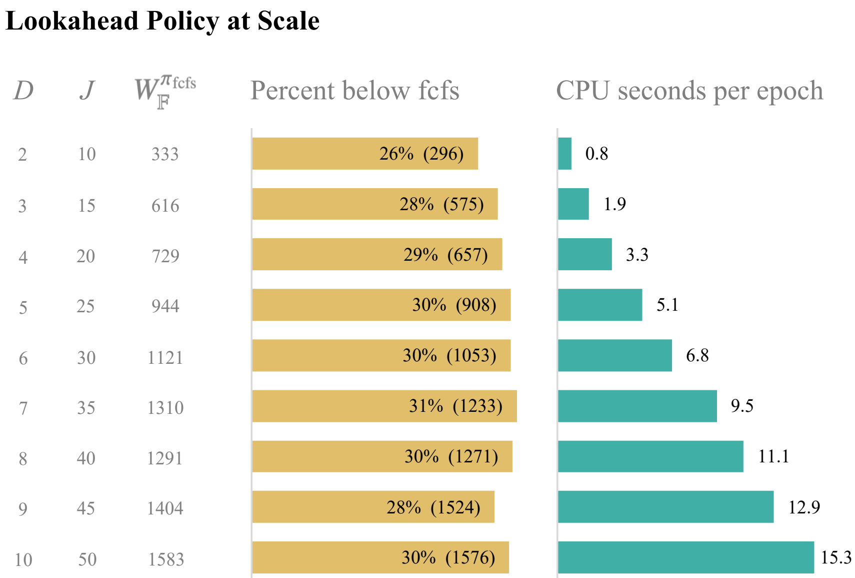
Experiments are conducted on a Linux machine equipped with an AMD EPYC 7453 processor and 1 TB of RAM. The processor consists of 112 cores clocked at 2.75 GHz. The lookahead policy is implemented in Java 21.0.2 and executed using a single thread. MILPs required for dual bound calculations are solved to optimality with CPLEX version 22.1.1.

6.3.1. Policy Quality and Dual Bounds Figure 2 shows dual bound $W^{\star}_{\mathbb{P}}$, the expected cost $\mathbb{E}[W^{\pi_{\text{one}}}_{\mathbb{G}}|\mathbb{F}]$ of the lookahead policy under the distribution filtration, the cost $W^{\pi_{\text{one}}}_{\mathbb{F}}$ of the lookahead policy under the natural filtration, and the cost $W^{\pi_{\text{fcfs}}}_{\mathbb{F}}$ of the first-come-first-served policy under the natural filtration, all as a percentage increase over the EVPI. The left portion of the figure shows values for problems with $D = 1$ dock and $J = 10$ trucks. The right portion increases the number of docks to $D = 2$. Tractability issues prevent calculation of the dual bound for larger instances.

Figure 2 shows that dual bound $W^{\star}_{\mathbb{P}}$ is strong. Recall from §5.2 that $W^{\star}_{\mathbb{P}}$ can be no larger than the expected cost $\mathbb{E}[\tilde{W}^{\star}_{\mathbb{G}}|\mathbb{F}]$ of an optimal MDP policy. Additionally, notice that $\mathbb{E}[W^{\pi_{\text{one}}}_{\mathbb{G}}|\mathbb{F}]$ must be at least as large as $W^{\star}_{\mathbb{P}}$. Because the gaps between $W^{\star}_{\mathbb{P}}$ and $\mathbb{E}[W^{\pi_{\text{one}}}_{\mathbb{G}}|\mathbb{F}]$ in Figure 2 are nearly zero, there is little room to improve $W^{\star}_{\mathbb{P}}$ for these instances. Further, compared to the EVPI, which is relatively straightforward to obtain, the more complex $W^{\star}_{\mathbb{P}}$ is 11 percentage points higher when $D = 1$ and 9 percentage points higher when $D = 2$. Not only is $W^{\star}_{\mathbb{P}}$ a better gauge for policy quality than the EVPI, but it is nearly the largest dual bound that can be obtained from the analysis of §5.

Figure 2 indicates that the lookahead policy is very good. When $D = 1$ the gap is 0.26 percent, and when $D = 2$, the gap is 0.62 percent. Thus, the policies are effectively optimal under the

Figure 3 Improvement over Practice and Computational Effort



distribution filtration. Under the natural filtration, the gaps are larger. When $D = 1$ policy cost $W_{\mathbb{F}}^{\pi_{\text{one}}}$ is 2 percentage points larger than $W_{\mathbb{P}}^{\star}$, and when $D = 2$ it is 10 percentage points higher. Because the dual bounds are virtually at the largest possible values, these gaps must be due to partial observability, suboptimal decisions, or both. The policy's strong performance under the distribution filtration suggests that the gaps under the natural filtration are primarily due to partial observability. The evidence points to the lookahead policy as making nearly optimal decisions, but because it must work with belief distributions on truck arrival times instead of with actual distributions, the policy costs are higher, and hence the gaps are larger.

Additionally, relative to the first-come-first-served policy common in practice, the lookahead policy is substantially better. When $D = 1$, the cost of the lookahead policy is 29 percentage points lower than the cost of the first-come-first-served policy. When $D = 2$, the difference is 15 percentage points. As we show= in the following, this trend holds as problem size grows. In these cases, even though the dual bound is computationally prohibitive to calculate, the results presented in Figure 2 give us confidence that the lookahead policy makes high quality decisions.

6.3.2. Policy Performance at Scale Figure 3 explores the performance of the lookahead policy at scale. For various numbers of docks D and trucks J, the figure shows the cost $W_{\mathbb{F}}^{\pi_{\text{one}}}$ of the lookahead policy as the percentage below the cost $W_{\mathbb{F}}^{\pi_{\text{fcfs}}}$ of the first-come-first-served policy. It also shows the average CPU time required to execute the lookahead decision rule in each epoch.

Figure 3 demonstrates that the lookahead policy is an attractive alternative to practice. Across all instances, the lookahead policy outperforms the first-come-first-served policy by a wide margin.

The average decrease in cost of $W_{\mathbb{F}}^{\pi_{\text{one}}}$ relative to $W_{\mathbb{F}}^{\pi_{\text{fcfs}}}$ is 29 percent. While we do not know how the lookahead policy compares to an optimal policy, the results in Figure 2 suggest that it is competitive. Additionally, the per-epoch CPU time is small enough for real-time decisions. Even at $D = 10$ docks and $J = 50$ trucks, which represents the high end of what might be encountered in practice, the lookahead policy requires only 15.3 seconds on average to return an action.

6.3.3. The Value of Filtering The literature on truck scheduling does not consider how the noise surrounding ETAs can impact decisions. As described in §2, the models and methods in the extant literature treat ETAs as actual truck arrival times. They do not employ Bayesian filters that use ETAs to update belief distributions. As we show in what follows, explicit consideration of ETAs as a noisy signal for arrival times is an important ingredient for better decisions.

We explore the benefit of filtering by comparing the lookahead policy to two methods that do not use filters. The first method modifies the lookahead policy to operate without belief distributions. In Equation (2), rather than consider a range of outcomes to estimate the cost-to-go, the decision rule considers the sole outcome that trucks en route arrive per the current ETAs. If e_j is the most recent ETA for truck j, then $\hat{a}_j = e_j$ for each truck $j \in \mathcal{J}^d(s_k)$ en route to the warehouse and $\hat{a}_j$ equals known arrival time a_j for each truck $j \in \mathcal{J} \setminus \mathcal{J}^d(s_k)$ already arrived to the warehouse. This event is assumed to happen with probability one, which yields the decision rule $\arg\min\{W_k(s_k, x_k) + \hat{W}(s_k^x, \hat{a}) : x_k \in \mathcal{X}(s_k)\}$. The second method mimics the rolling horizon procedures that dominate the truck scheduling literature described in §2. The ILS is executed from current state s_k with arrival times $\hat{a}$ and returns order $\hat{\gamma}$. Let $\hat{g} = \min\{g \in \{1, \ldots, |\hat{\gamma}|\} : u_{\hat{\gamma}(g)} = t_k\}$ be the smallest index of $\hat{\gamma}$ such that the service start time is equal to current time t_k. Then, $\hat{\gamma}(\hat{g})$ is the first truck in order $\hat{\gamma}$ to start service at time t_k. If $\hat{\gamma}(\hat{g})$ exists, the decision rule assigns truck $\hat{\gamma}(\hat{g})$ to an available dock. If $\hat{\gamma}(\hat{g})$ does not exist, the decision rule returns the zero action.

Figure 4 shows the cost of the lookahead policy without filtering and the cost of the rolling horizon policy without filtering, both as percent above the cost of the lookahead policy with filtering. The analysis uses the same problem instances associated with Figure 3. The figure underscores the vital role of filtering. Without filtering, the average cost of the lookahead policy is 8.7 percent higher. Notably, the rolling horizon policy without filtering is 9 percent higher, on average, than the lookahead policy with filtering. Thus, when belief distributions are ignored, there is little benefit to using the lookahead mechanism beyond what is known in the current state. To substantially improve decisions, uncertainty surrounding ETAs must be explicitly incorporated into action selection.

Figure 4 To Filter, or Not to Filter

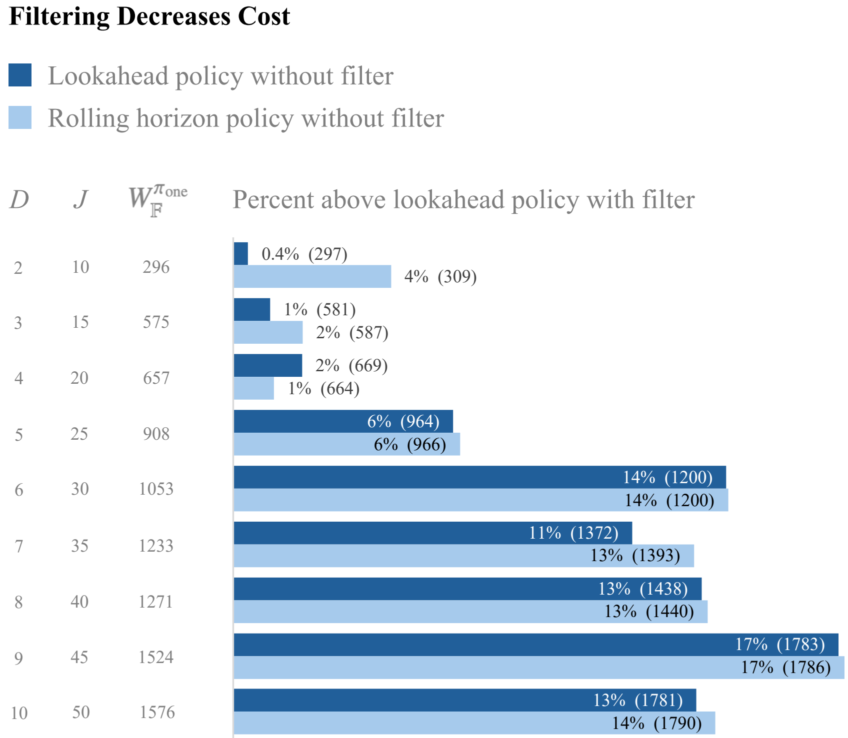
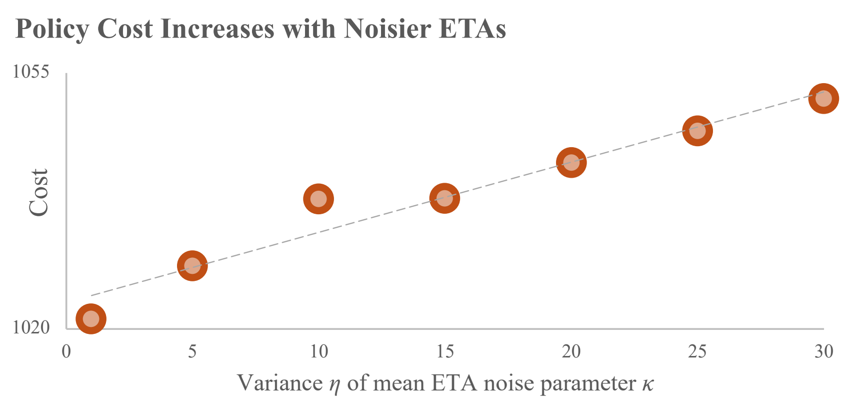


Figure 5 Lookahead Policy Cost vs. ETA Quality



6.3.4. ETA Accuracy Figure 5 examines the impact of ETA accuracy. Up to this point, the variance η of the mean ETA noise parameter κ has been set to 1. The figure presents the costs of lookahead policies when η is increased to 5, 10, 15, 20, 25, and 30. The values are averages across the same problem instances associated with Figure 3.

Expectedly, Figure 5 demonstrates that more accurate ETAs lead to lower costs than less accurate ETAs. Increasing η from 1 to 30 leads to a cost increase of almost 3 percent. However, even when η is at 30, the average cost of the lookahead policy is substantially lower than the average cost of the first-come-first-served policy when η equals 1. Thus, our methodology is still valuable even when the noise to signal ratio is high.

7. Conclusions

The demands placed on today's logistics networks are considerable. In an industry with thin margins, customers' expectations of short lead times and responsiveness are difficult to satisfy. Our lookahead policy for scheduling inbound trucks at warehouses and distribution centers significantly reduces expected truck waiting times relative to current practice, on average by 29 percent. Not only can these reductions improve customer satisfaction across the network, but they can also have a meaningful impact on profitability.

Beyond potential contributions to the practice of logistics, our work contributes to transportation research in two notable ways. First, we recognize that inbound truck arrival time distributions are often difficult to observe. However, we show how to approximate them with ETAs. Incorporation of ETAs into our POMDP more accurately models the actual information available to dispatchers. Moreover, in contrast to the truck scheduling literature, which takes ETAs as point estimates, we show that filtering ETAs to characterize uncertainty in truck arrival times is key to better scheduling decisions.

Second, we use information relaxations and an information penalty to develop a dual bound that is much stronger than a bound obtained only through perfect information. On average, our penalized dual bound is nearly 10 percent larger than the expected value with perfect information. With a duality gap of less than one percent, our bound allows us to state with confidence that the lookahead policy is nearly optimal when truck arrival time distributions are fully observed. When distributions are hidden, this result gives credibility to the notion that a larger duality gap of about 10 percent is due primarily to partial observability of the arrival time distributions rather than suboptimality of the policy. Although our analyses are specific to inbound truck scheduling with ETAs, they serve as a template for the transportation science and logistics community to develop dual bounds for other problems.

Future work on dynamic and stochastic inbound truck scheduling may need to navigate the tension between realism and tractability. For example, a more true-to-life model might incorporate service time uncertainty. Although such an extension seems pragmatic, it may pose substantial challenges. Certainly heuristic policies can be developed for such a problem. But proving their goodness via dual bounds may require substantial investment. For now, adding ETAs to the collection of models and methods for dynamic truck scheduling is a significant step forward.

Appendix

A. Iterated Local Search

Algorithm 1 details the ILS procedure discussed in §4.1. Line 1 initializes the process. The search begins with an order γ_{fcfs} that sequences trucks first-come-first-served by their arrival times $\hat{a}$. Order γ_{fcfs} initializes the incumbent solution $\hat{\gamma}$. Iteration counter i begins at 0. The loop on line 2 executes perturbation followed by local search $I_{\max} = 5000$ times. Lines 3–7 comprise the perturbation phase. Every $P = 250$ iterations, perturbation parameter r is set to $r_1 = 0.7$. Otherwise, r is set to $r_2 = 0.3$. Order γ is established by randomly relocating $\lfloor r \cdot |\mathcal{J}^d(s) \cup \mathcal{J}^w(s)| \rfloor$ elements of incumbent solution $\hat{\gamma}$. This operation relocates up to a proportion r of the the number of trucks en route to the warehouse and the number of trucks waiting at the warehouse for assignment to a dock. Lines 8–14 comprise the local search phase. The loop on line 8 executes first-improving local search across the four neighborhood structures listed on line 9. Whenever an order γ' is found to improve on order γ, lines 10–13 replace γ with γ' and reset the search. The process iterates until an improving solution cannot be found. If the cost $\hat{W}(s, \hat{a}, \gamma)$ of the resulting order γ is smaller than the cost $\hat{W}(s, \hat{a}, \hat{\gamma})$ of the incumbent solution $\hat{\gamma}$, then lines 15 and 16 replace $\hat{\gamma}$ with γ. Line 17 increments the iteration counter. The search concludes on line 18.

B. Lookahead Validation

The lookahead mechanism described in §4 is the basis for the lookahead policy. In this section, we validate the mechanism and explore the proposed acceleration procedures for decision rule execution.

The lookahead mechanism employs ILS to heuristically solve the optimization problem $\min\{\hat{W}(s, \hat{a}, \gamma) : \gamma \in \Gamma(s)\}$. The aim of the ILS is to quickly identify high quality solutions. To verify that the ILS performs well, we compare heuristic solution values returned by the ILS to solution values obtained through mixed integer linear programming. We also track the time required to obtain each. We take initial state s_0 as the input state. For a given sequence of truck arrival times $\hat{a}$, the optimization problem treated by the ILS is equivalent to the perfect information problem with zero penalty described in §6.2. We use the CPLEX solver to address the perfect information MILP.

Table 2 presents objective values and computing times across a range of problem instances. For each combination of docks D and trucks J, we generate service times as described in §6, plus 150 sequences of truck arrival times $\hat{a}$ from the unobserved arrival time distributions. The figures

Algorithm 1: Iterated Local Search

Input: State s and arrival times $\hat{a}$

Output: Order $\hat{\gamma}$

$\hat{\gamma} \leftarrow \gamma_{\text{fcfs}}$, $i \leftarrow 0$
while $i < I_{\max}$ **do**
    if $i \bmod P = 0$ **then**
        $r \leftarrow r_1$
    else
        $r \leftarrow r_2$
    $\gamma \leftarrow$ perturb $\lfloor r \cdot |\mathcal{J}^d(s) \cup \mathcal{J}^w(s)| \rfloor$ trucks in $\hat{\gamma}$
    repeat
        foreach neighborhood $N \in \{$two-way swap, insertion, four-way swap, insertion and swap$\}$ **do**
            Find first $\gamma' \in N(\gamma)$ such that $\hat{W}(s, \hat{a}, \gamma') < \hat{W}(s, \hat{a}, \gamma)$
            if such γ' exists **then**
                $\gamma \leftarrow \gamma'$
                break
    until no improvement
    if $\hat{W}(s, \hat{a}, \gamma) < \hat{W}(s, \hat{a}, \hat{\gamma})$ **then**
        $\gamma^{\star} \leftarrow \gamma$
    $i \leftarrow i + 1$
return $\hat{\gamma}$

reported in the table are averages across the 150 optimization problems corresponding to these samples. Columns three, four, and five report the best objective value obtained by CPLEX, the number of CPU seconds required to obtain the value, and the number of instances out of 150 for which CPLEX identifies an optimal solution, respectively. The CPU time is capped at two hours. Columns six, seven, and eight report analogous figures for the ILS. Column nine indicates the number of arrival time realizations out of 150 for which the ILS solution value is strictly less than the solution value obtained by CPLEX. The results in Table 2 stop at problem instances with $D = 5$ docks and $J = 25$ trucks because CPLEX solution times far exceed the two-hour cap for larger instances.

Table 2 confirms that the ILS solutions are high quality. When D equals 2 and 3, CPLEX finds the optimal solution value within the allotted time. The ILS matches these values. When D equals 4 and 5, CPLEX does not always identify an optimal solution. In these cases, on average the ILS identifies better solutions. Moreover, across all instances, the CPU time required by the ILS procedure is smaller than the CPU time required by CPLEX. When D is 3 and larger, the difference

Table 2 ILS: High Quality Solutions and Fast Computation

		CPLEX			ILS			
D	J	Objective	CPU	#Optimal	Objective	CPU	#Optimal	#Better
2	10	245.1	1.3	150	245.1	0.4	150	0
3	15	432.0	5957.7	150	432.0	0.9	150	0
4	20	463.1	7051.6	95	462.6	1.9	76	52
5	25	651.6	7203.2	56	651.2	3.5	40	88

CPLEX vs. ILS objective values and CPU times (seconds).

is considerable. These results validate the ILS as a means of obtaining high quality solutions to the lookahead optimization.

Although the ILS executes quickly, because it must be executed many times to evaluate decision rules, the procedure can become computationally burdensome as problem size grows. The acceleration procedures proposed in §4 provide four ways to decrease the time required to execute a decision rule. Following the order of presentation in §4, we label the methods as (i), (ii), (iii), and (iv). Table 3 shows the impact of the four rules on policy quality and computation time. For lookahead policy π_{one}, the table presents policy costs and CPU seconds per epoch for a range of problem instances. Each figure is an average across instance realizations as described in §6. Columns correspond to results obtained through none of the methods, each method in isolation, and all the methods in tandem.

The figures in Table 3 indicate that computation time can be significantly reduced with only nominal loss in policy quality. Relative to the case of using none of the methods, each method in isolation offers some improvement in CPU seconds per epoch. However, using all methods in tandem dramatically reduces computation time, on average by 84 percent. The reduction is large enough to facilitate the use of lookahead decision rules for real-time decisions. Moreover, the average increase in policy cost is less than 2 percent.

Acknowledgments

This research is part of a broader project financed by the European Union NextGenerationEU (National Sustainable Mobility Center CN00000023, Italian Ministry of University and Research Decree n.1033–17/06/2022, Spoke 10 “Sustainable Logistics”). We would like to thank all partners and companies that have actively contributed to making this research possible.

References

Brown, D. B. and Smith, J. E. (2022). Information relaxations and duality in stochastic dynamic programs: A review and tutorial. Foundations and Trends® in Optimization, 5(3):246–339.

Table 3 Lookahead Policy Speed-Ups

		Cost						CPU					
D	J	None	(i)	(ii)	(iii)	(iv)	All	None	(i)	(ii)	(iii)	(iv)	All
2	10	295.7	295.9	294.5	295.1	295.1	295.9	1.3	0.9	1.3	1.3	1.3	0.8
3	15	564.9	565.6	574.1	566.7	564.8	575.5	5.9	3.1	4.4	5.9	5.1	1.9
4	20	638.0	642.4	654.3	638.3	639.8	656.5	18.0	7.5	10.3	17.9	15.5	3.3
5	25	886.2	888.1	916.3	889.3	888.2	907.9	47.6	17.6	23.3	47.4	35.9	5.1
6	30	1025.1	1026.2	1060.6	1027.2	1022.7	1053.4	102.6	32.3	42.6	102.5	74.6	6.8
7	35	1209.9	1222.7	1250.3	1207.6	1207.4	1233.4	220.7	58.6	80.9	220.7	151.6	9.5
8	40	1256.6	1320.8	1258.7	1252.0	1276.1	1270.7	216.7	121.7	216.1	248.8	174.6	11.1
9	45	1489.2	1645.2	1495.3	1509.2	1546.7	1523.5	353.9	121.6	355.5	224.0	185.2	12.9
10	50	1549.0	1674.3	1585.0	1578.9	1599.1	1575.8	369.8	178.5	327.5	331.7	197.4	15.3

For one-step lookahead policy π_{one}, the cost and CPU seconds per epoch under each of four speed-up methods, under none of the methods, and under all of the methods.

Brown, D. B., Smith, J. E., and Sun, P. (2010). Information relaxations and duality in stochastic dynamic programs. Operations Research, 58(4-part-1):785–801.

Cekała, T., Telec, Z., and Trawiński, B. (2015). Truck loading schedule optimization using genetic algorithm for yard management. In Nguyen, N. T., Trawiński, B., and Kosala, R., editors, Intelligent Information and Database Systems, pages 536–548. Springer, Cham, Switzerland.

Chou, M. C., Liu, H., Queyranne, M., and Simchi-Levi, D. (2006). On the asymptotic optimality of a simple on-line algorithm for the stochastic single-machine weighted completion time problem and its extensions. Operations Research, 54(3):464–474.

Grand View Research (2023). Warehousing market size, share & trends report. `https://www.grandviewresearch.com/industry-analysis/warehousing-market-report`. Accessed: December 19, 2024.

Gupta, V., Moseley, B., Uetz, M., and Xie, Q. (2020). Greed works—online algorithms for unrelated machine stochastic scheduling. Mathematics of Operations Research, 45(2):497–516.

Heidari, F., Zegordi, S. H., and Tavakkoli-Moghaddam, R. (2018). Modeling truck scheduling problem at a cross-dock facility through a bi-objective bi-level optimization approach. Journal of Intelligent Manufacturing, 29(5):1155–1170.

Heydar, M., Mardaneh, E., and Loxton, R. (2022). Approximate dynamic programming for an energy-efficient parallel machine scheduling problem. European Journal of Operational Research, 302(1):363–380.

Kanet, J. J. and Sridharan, V. (2000). Scheduling with inserted idle time: Problem taxonomy and literature review. Operations Research, 48(1):99–110.

Kleywegt, A. J., Shapiro, A., and Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization, 12(2):479–502.

Konur, D. and Golias, M. M. (2013). Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach. Transportation Research Part E: Logistics and Transportation Review, 49(1):71–91.

Larbi, R., Alpan, G., Baptiste, P., and Penz, B. (2011). Scheduling cross docking operations under full, partial and no information on inbound arrivals. Computers & Operations Research, 38(6):889–900.

Liu, X., Chu, F., Zheng, F., Chu, C., and Liu, M. (2021). Parallel machine scheduling with stochastic release times and processing times. International Journal of Production Research, 59(20):6327–6346.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2003). Iterated local search. In Glover, F. and Kochenberger, G. A., editors, Handbook of Metaheuristics, pages 320–353. Springer, Boston, MA.

Maecker, S., Shen, L., and Mönch, L. (2023). Unrelated parallel machine scheduling with eligibility constraints and delivery times to minimize total weighted tardiness. Computers & Operations Research, 149:105999.

Megow, N., Uetz, M., and Vredeveld, T. (2006). Models and algorithms for stochastic online scheduling. Mathematics of Operations Research, 31(3):513–525.

Mejía, G., Sarmiento, A., and Guerrero, W. (2023). Truck and cross-docking scheduling in the flower export sector in colombia. Transportation Research Record, 2677(5):352–366.

Modica, T., Tappia, E., Colicchia, C., and Melacini, M. (2024). Integrating arrival time estimation in truck scheduling: an explorative study in grocery retailing. Production & Manufacturing Research, 12(1):2425678.

Nasiri, M. M., Ahmadi, N., Konur, D., and Rahbari, A. (2022). A predictive-reactive cross-dock rescheduling system under truck arrival uncertainty. Expert Systems with Applications, 188:115986.

Pinedo, M. L. (2022). Scheduling: Theory, Algorithms, and Systems. Springer, Cham, Switzerland, 6th edition.

Ronconi, D. P. and Powell, W. B. (2010). Minimizing total tardiness in a stochastic single machine scheduling problem using approximate dynamic programming. Journal of Scheduling, 13(6):597–607.

Särkkä, S. (2013). Bayesian Filtering and Smoothing. Institute of Mathematical Statistics Textbooks. Cambridge University Press.

Simon, D. and Simon, D. L. (2010). Constrained Kalman filtering via density function truncation for turbofan engine health estimation. International Journal of Systems Science, 41(2):159–171.

Song, R., Lau, H. C., Luo, X., and Zhao, L. (2022). Coordinated delivery to shopping malls with limited docking capacity. Transportation Science, 56(2):501–527.

Vanga, R., Maknoon, Y., Tavasszy, L. A., and Gelper, S. (2022). Effect of real-time truck arrival information on the resilience of slot management systems. In 2022 Winter Simulation Conference (WSC), pages 1593–1602. IEEE.

Xi, X., Changchun, L., Yuan, W., and Loo Hay, L. (2020). Two-stage conflict robust optimization models for cross-dock truck scheduling problem under uncertainty. Transportation Research Part E: Logistics and Transportation Review, 144:102123.

Xu, B., Liu, X., Li, J., Yang, Y., Wu, J., Shen, Y., and Zhou, Y. (2022). Dynamic appointment rescheduling of trucks under uncertainty of arrival time. Journal of Marine Science and Engineering, 10(5):695.

Yu, V. F., Sharma, D., and Murty, K. G. (2008). Door allocations to origins and destinations at less-than-truckload trucking terminals. Journal of Industrial and Systems Engineering, 2(1):1–15.

Zhang, Z., Zheng, L., Li, N., Wang, W., Zhong, S., and Hu, K. (2012). Minimizing mean weighted tardiness in unrelated parallel machine scheduling with reinforcement learning. Computers & Operations Research, 39(7):1315–1324.