A Rollout Algorithm for Truck Scheduling Using Estimated Time of Arrival

Maximiliano Cubillos*, **Ola Jabali, Elena Tappia**
Politecnico di Milano, Milano, Italy
*Email: maximiliano.cubillos@polimi.it

Justin Goodson
Saint Louis University, St. Louis, USA

1 Introduction

Yard management consists of optimization problems related to the synchronization of truck arrivals, departures, and resource utilization within logistic distribution centers. We consider minimizing the total waiting time of inbound trucks at a yard by optimizing the truck-to-dock assignment decisions. The inherent uncertainty of truck arrival times and their impact on assignment decisions can be mitigated by the use of technologies that provide estimated time of arrivals (ETA) in quasi-real-time. The aim of this paper is to propose a dynamic algorithm that accounts for ETA information to make online truck assignment decisions.

We consider a set $\mathcal{J}$ of n inbound trucks to be assigned to a set $\mathcal{I}$ of m docks in a distribution center. Upon arrival, trucks can be directly assigned to start processing at a dock, or to a waiting yard. Each truck has a known processing time p_j, and we assume processing at a dock is non-preemtive. The arrival time of each truck follows a distribution which is never observed by the decision maker. Instead, the decision maker has a *belief distribution* for each truck, which is periodically updated based on ETA information using a Bayesian filter. We formulate the resulting problem as a Markov Decision Process (MDP) with the objective of finding a policy that minimizes the total waiting time of all trucks. This is defined as the difference between the moment the truck starts being processed and the time it arrives to the yard. The main contributions of this study are as follows: (1) we propose an MDP model with tailored decision epochs and states to address the integration of the ETAs into the scheduling decisions; (2) to overcome the explosion in the dimension of the state and action spaces, we propose a heuristic solution method based on a rollout algorithm with an embedded Iterated Local Search (ILS) heuristic; (3) aside from a perfect information bound, we propose a penalized lower bound based on an information relaxation scheme; and (4) we demonstrate the effectiveness of our algorithm compared to the proposed lower bounds and benchmark scheduling policies, based on generated test instances.

2 MDP formulation

We assume a decision epoch k is triggered by the first event among: (1) the arrival of a truck, (2) a truck finishes its process, i.e., a dock becomes available, and (3) *schedule verification* epochs. A schedule verification epoch is triggered when reaching an expected next arrival of a truck, before an actual truck arrival. An action x_k is a feasible assignment of trucks to docks from the set of trucks that are either at the waiting yard or just arrived. Based on the action and the possible decision epoch triggers, a state s_k transits from a post-decision s_k^x to a pre-decision state s_{k+1}, updating belief distributions based on the ETA information. We define a waiting time function W_k as the expected difference between the current and the next decision epoch times:

$$W_k(s_k, x_k) = \sum_{j \in \mathcal{J}_k^w} \mathbb{E}[t_{k+1} - t_k \mid s_k, x_k],$$

where t_k is the time and $\mathcal{J}_k^w$ is the set of trucks in the waiting yard in decision epoch k. Since the next decision epoch time t_{k+1} is unknown at t_k, the expectation is taken with respect to the belief arrival distribution of the incoming trucks. The objective is to find a policy π in the set of feasible policies Π, defined as a sequence of truck-to-dock assignments, that minimizes the expected value of the waiting times from the initial state until a terminal state K:

$$V(\pi^*) = \min_{\pi \in \Pi} \mathbb{E} \left[\sum_{k=0}^{K} W_k(s_k, x_k^\pi(s_k)) \Big| s_0 \right].$$

3 Rollout algorithm

A rollout algorithm is an adaptive online forward dynamic programming mechanism that makes real-time decisions for realized states by employing a lookahead decision rule. Following the early work of Bertsekas et al. [1], this methodology has been successfully applied in different dynamic applications [2]. The algorithm operates within a loop, which continues until the current state reaches a terminal state K. Within each iteration, a decision rule is evaluated based on the current states to determine the best action. We propose a decision rule that looks one decision epoch ahead in the future for all possible post-decision states s_k^x, generates heuristic policies $\pi(s_{k+1})$ based on the heuristic algorithm $\mathcal{H}$ for the set of all possible future states, and selects the action x_k that minimizes the expected waiting time based on $\pi(s_{k+1})$. The set of possible future states is based on a competition between the arrival of all incoming trucks, where each event represents the case of one of the incoming trucks arriving first, or a dock becoming available first. A subset of the heuristic policy input and the probability of reaching the state s_{k+1} are calculated based on the belief distributions. Both the probabilities and the expected arrival times used in $\mathcal{H}$ require conditional distributions on the minimum of the arrival distributions, which are intractable. To overcome this, we estimate their values using simulation.

The heuristic $\mathcal{H}$ solves the underlying deterministic scheduling problem, which is equivalent to the Parallel Machine Scheduling Problem with release dates (PMSPR). To solve the PMSPR, we propose an ILS heuristic that uses as input the expected arrival times of the incoming trucks given a reachable state s_{k+1}. The ILS uses the sequence of trucks based on the start service times of a solution, iterates feasible solutions applying a local search to produce local optima, and diversifies the solution. The algorithm starts with an initial solution, given by a First Come First Serve (FCFS) rule, which is improved through repeated iterations of a Variable Neighbourhood Descent (VND) search and perturbation phases. In the VND, we use four neighborhood structures based on swap and reinsertion moves in the sequence of trucks.

4 Lower bounds

In order to assess the performance of our algorithm, we propose two lower bounds based on information relaxations of the problem. First, we propose a bound in which we assume actual arrival times to be known at the beginning of the planning horizon for all trucks. We refer to this problem as the perfect information bound (PIB). The resulting problem is equivalent to the PMSPR. We obtain the PIB by proposing a sequence-based MILP. Second, similar to Brown et al. [3], we propose a tighter bound in which we assume the *distribution* of arrival times are known, while the actual arrivals are unknown. We refer to this problem as the penalized bound (PB). In the PB, we select a penalty function that compensates for the distribution information and derive a MILP formulation based on the original MDP formulation.

5 Results and discussion

We compare the expected objective value of the rollout policy with the PIB, PB, and the FCFS and the Shortest Processing Time First (SPTF) rules as a benchmark policies. For this, we compute the average over 10 realizations of arrival times for generated test instances, where each instance corresponds to one realization of processing times from a uniform distribution $p_j \sim \mathcal{U}(1, 20)$. We assume the actual arrival times are normally distributed, with mean parameters generated from a uniform distribution in the range $(0, \mathcal{P})$, such that $\mathcal{P} = \sum_{j \in \mathcal{J}} p_j / m$, and the variance parameters are set to 2. For each instance, we compute the gap between the objective function obtained by the tested policies using gap(%) $= 100 \frac{V_{\text{PB}} - V_x}{V_{\text{PB}}}$, where V_{PB} and V_x are the objective values of the PB and the given policy, respectively. The experiments were conducted on a computer with a 3.1

GHz Dual-Core Intel Core i5 with 16 GB of RAM.

In Table 1 we summarize preliminary results for test instances using $m = 2$ and $n = 10$, setting the belief distributions equal to the actual distributions. First, we observe that the PB provides a tighter lower bound compared to the PIB, increasing the average objective value from 39.6 to 42.2. Second, we observe that the rollout algorithm obtains lower or equal gaps compared to the benchmark policies in all cases, with an average gap of 2.5% compared to 23.9% and 9.4% for FCFS and SPTF, respectively. We will further evaluate the effect of the dynamics on generating realistic ETA update scenarios.

	PIB		PB		FCFS		SPTF		Rollout		
Inst.	Obj.	Runtime (s)	Obj.	Time (s)	Obj.	Gap(%)	Obj.	Gap(%)	Obj.	Gap(%)	Time (s)
1	49.1	2.1	52.7	26.1	61.5	16.8	54.4	3.3	53.7	2.0	70.5
2	83.5	4.4	89.4	76.1	113.6	27.1	89.4	0.0	89.4	0.0	70.2
3	28.3	0.4	29.4	2.3	41.4	40.9	38.3	30.2	30.9	5.2	55.0
4	8.8	0.5	9.1	2.0	10.1	10.2	10.1	10.3	9.5	4.3	66.4
5	28.3	0.5	30.6	4.0	38.0	24.1	31.6	3.3	30.9	1.0	64.0

Table 1: Comparison of the rollout algorithm with the PIB, PB, and benchmark policies.

References

[1] D. Bertsekas, J. Tsitsiklis and C. Wu, "Rollout algorithms for combinatorial optimization", *Journal of Heuristics* 3 (3), 245–262, 1997.

[2] J. Goodson, B. Thomas, and J. Ohlmann, "A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs", *European Journal of Operational Research* 258(1), 216–229, 2017.

[3] D. Brown, J. Smith and P. Sun, "Information relaxations and duality in stochastic dynamic programs", *Operations Research* 58(4), 785–801, 2010.