Received: 4 March 2016 | Revised: 24 June 2018 | Accepted: 27 June 2018

DOI: 10.1002/nav.21804

Check for updates

RESEARCH ARTICLE

WILEY

Aggregating courier deliveries

Patrick R. Steele | **Shane G. Henderson** | **David B. Shmoys**

Operations Research and Information Engineering, Cornell University, Ithaca, New York

Correspondence
Patrick R. Steele, Operations Research and Information Engineering, Cornell University, Ithaca, NY
Email: prs233@cornell.edu

Funding information
National Science Foundation, CCF-1522054, CCF-1526067, CMMI-1200315, CMMI-1537394.

Abstract

We consider the problem of efficiently scheduling deliveries by an uncapacitated courier from a central location under online arrivals. We consider both adversary-controlled and Poisson arrival processes. In the adversarial setting we provide a randomized $(3\beta\Delta/2\delta - 1)$-competitive algorithm, where β is the approximation ratio of the traveling salesman problem, δ is the minimum distance between the central location and any customer, and Δ is the length of the optimal traveling salesman tour overall customer locations and the central location. We provide instances showing that this analysis is tight. We also prove a $1 + 0.271\Delta/\delta$ lower-bound on the competitive ratio of any algorithm in this setting. In the Poisson setting, we relax our assumption of deterministic travel times by assuming that travel times are distributed with a mean equal to the excursion length. We prove that optimal policies in this setting follow a threshold structure and describe this structure. For the half-line metric space we bound the performance of the randomized algorithm in the Poisson setting, and show through numerical experiments that the performance of the algorithm is often much better than this bound.

KEYWORDS

approximation algorithms, competitive analysis, courier, Markov decision process

1 | INTRODUCTION

Many companies are augmenting traditional centralized logistics planning by offering on-demand delivery services. Uber Rush and Amazon Prime Now have recently begun offering on-demand delivery services of online purchases with the goal of delivering the package within 1 hour. Orders for these services arrive in an online fashion and so delivery routes need to be updated in real-time. Additionally, each of these services utilize local couriers to complete the deliveries (Amazon flex, 2005; Uber rush, 2005). Independent contractors are interested in utilizing their time efficiently while still performing all deliveries in a timely manner. Since couriers operate in urban areas where orders can originate from clustered retailers, there is the possibility that multiple independent orders can be fulfilled simultaneously. Thus there is a tension between immediately embarking on a delivery trip when an order comes in and waiting some amount of time for additional jobs to arrive that can be batched in with existing orders.

How should a courier decide when to make a delivery immediately and when to wait? There are at least two scenarios of interest to consider, both of which depend on how the company dispatching the orders assigns jobs to couriers. First, the company might try to allocate orders to couriers in a round-robin fashion. In this case a courier receiving an order might not expect another order for a while and so might depart immediately. Alternatively, the company might assign orders myopically to the nearest courier. In this case the courier knows after receiving an order that subsequent orders to the same location might also be assigned to him; in this case it might wait for additional orders before departing.

To explore this we consider a system consisting of a retailer serving orders to customers in a metric space from a single depot using a single uncapacitated courier. Each order consists of a customer requesting a unique good to be moved from the retailer's depot to the customer's location in the metric space by the courier. The assumption of unique goods prevents a good marked for one customer from being used to serve another. Orders are placed over time, and the retailer

 © 2018 Wiley Periodicals, Inc.

only gains knowledge of an order when it is placed. An order can be picked up by the courier as soon as it arrives but not before. This prevents the courier from moving goods for customers that have not yet placed an order. When the courier arrives at a customer with the associated good the order is immediately completed. The goal is to minimize the total latency of all orders, where the latency of an order is the time between the order being placed and the associated good being delivered to the associated customer. This objective function is chosen to offer the customers prompt service. Other objectives such as minimizing the latest completion time are heavily influenced by the times that orders arrive in the system, making the completion times for early orders unimportant.

We consider the problem from two perspectives. First, we consider the case where the time and location of arrivals are controlled by an adversary. We present a randomized algorithm with a competitive ratio of $3\beta\Delta/2\delta - 1$, where β is the approximation ratio of the traveling salesman problem (TSP) (Christofides, 1976a), δ is the minimum distance between the depot and any customer, and Δ is the length of the optimal TSP tour over all locations in a finite metric space $\mathcal{S}$. In this setting we also provide a lower bound on the competitive ratio of any algorithm of $1 + 0.271\Delta/\delta$. Second, we consider the case where orders occur according to a Poisson process. Here we assume that travel times are exponentially distributed rather than being deterministic, making the problem amenable to analysis. In this case we derive structural results on the optimal policies for serving such order sequences; in particular, we show that the optimal policy for such an arrival process is a threshold policy where the choice to depart with a collection of orders implies that we will also choose to depart when there are more orders to those same locations. Finally, we show through numerical experiments that the performance of the randomized competitive algorithm compares favorably with the optimal threshold policy in restricted settings.

1.1 | Literature review

The *vehicle routing problem* (VRP) considers the problem of minimizing the cost required to serve a set of customer demands with a fleet of vehicles. In the most basic setting all quantities are known in advance, and so in principle a master schedule may be computed before the day begins. Exact and heuristic solution techniques inspired by the TSP are considered by Christofides (1976b). A closely related problem is the *dial-a-ride problem* (DARP) that consists of choosing vehicle routes and schedules for a sequence of pickup and delivery requests; Cordeau and Laporte (2007) offer a survey of the DARP. Feuerstein and Stougie (2001) consider the DARP in the single server setting as we do. They assume a single (capacitated or uncapacitated) server must serve a sequence of pickup and delivery requests in a general metric space in an online fashion, and provide algorithms with constant competitive ratio. They also consider the special case of the metric space being the real line. Unlike our work, they consider minimizing both the latest completion time of all jobs and the average completion time of all jobs, while we consider minimizing the total latency of all jobs. Ascheuer, Krumke, and Rambau (2000) consider the online DARP and introduce algorithms with constant competitive ratio. As with, Feuerstein and Stougie (2001) our results differ in that we minimize the total latency of jobs rather than the maximum lateness.

The *TSP* is a classic optimization problem in which the goal is to compute a minimum cost tour over n cities in a metric space (Williamson & Shmoys, 2011). If we take edge costs to the time required to traverse this edge, the optimal TSP tour computes the minimum time required for a courier to depart from a depot, make a number of deliveries, and return to the depot, known as the makespan. The *online TSP* (OLTSP), introduced by Ausiello, Feuerstein, Leonardi, Stougie, and Talamo (2001) is a natural variant of the TSP where the cities to be visited are revealed over time. They give an algorithm that is two-competitive, along with competitive algorithms for the case where the courier does not end the tour at the depot. The problem we consider is closely related to the OLTSP in that the customers we need to visit are revealed over time as orders are placed, and differs in that we must return to the depot to pick up a customer order before serving that customer. More recently Jaillet and Xin (2014) consider the OLTSP with the option to reject new cities as they arrive, incurring some penalty for each rejected city. They provide a two-competitive algorithm for the case where cities can be rejected at any time, and show that this is the best possible competitive ratio. In the restricted case where the choice to accept or reject cities must be made when the city arrives in the problem they provide an $\Omega(\sqrt{\log n})$ lower-bound on the competitive ratio, where n is the number of cities.

The OLTSP problem can be viewed as a one-courier instance of a VRP where all the products being delivered are fungible and the courier departs the depot with an infinite supply of goods; this ensures that the courier does not need to return to the depot before all cities have been visited. Our objective is to minimize the total latency of orders, rather than minimizing the makespan of the delivery schedule.

If the VRP with a makespan objective is viewed as a generalization of the TSP, then the VRP with a total latency objective can be viewed as a generalization of the *TRP* (Krumke, de Paepe, Poensgen, & Stougie, 2003). Like the TSP, the input to the TRP is a set of cities in a metric space, and the output is a tour over those cities. However, where the TSP finds a tour to minimize the time it takes to return to the depot, the TRP finds a tour to minimize the sum of the times it takes to visit each individual city. In the case where all orders are placed at time zero our problem is exactly the TRP.

A natural extension of the TRP is the online TRP (OTRP) where cities to be served are revealed over time. Krumke et al. (2003) present a 4.3281-competitive algorithm, along with a lower bound of 7 of 3 on the competitive ratio of

any algorithm for the OTRP. We generalize these results to the setting where orders must be picked up from a central depot before being delivered to each city. Irani, Lu, and Regan (2004) consider the online dynamic traveling repairman problem (ODTRP), a variant of the OTRP where service requests occur over time and have deadlines. They consider serving requests in a metric space with a single courier, and seek to maximize the number of requests completed before the associated deadline. As a consequence of this objective function they can choose whether to serve a request or not, and have constraints on the amount of time the server is allowed to delay before committing to serving or abandoning a request. They provide competitive algorithms with competitive ratios bounded function of the diameter of the metric space, along with lower bounds on those ratios for deterministic algorithms. The dependence on the diameter of the metric space in their bounds on the competitive ratio is similar to our dependence on the length of the optimal TSP tour over the metric space, as these parameters are an intrinsic property of the metric space rather than a consequence of the allowed actions of an adversary. Recently Sitters (2014) provides a fully polynomial-time approximation scheme for the OTRP when the metric space is the Euclidean plane or weighed trees. The approach taken transfers to other problems, resulting in an improved approximation bound for the single-machine weighted sum of completion time scheduling problem with precedence constraints.

In section 2 we consider our problem in an online adversarial setting. We use the competitive analysis framework of Borodin and El-Yaniv (1998) to present and analyze the problem. In section 3 we consider our problem in the average-case setting. Here we use the tools of Sennott (2009) for our analysis. Sennot additionally provides the computational techniques we use to perform numerical experiments.

1.2 | Contributions

Our key contributions to the literature are providing a randomized algorithm to minimize the total latency of orders with a competitive ratio of $3\beta\Delta/2\delta - 1$, which is within a factor of 8.4 of the lower bound on the competitive ratio of $1 + 0.271\Delta/\delta$, using $\beta = 3/2$ as in Christofides (1976a). These results follow in the style of Krumke et al. (2003), where competitive ratios are given within a constant factor of the best possible ratios. Our results are in contrast to those of Feuerstein and Stougie (2001) which seek to minimize the average completion time. Additionally, Feuerstein and Stougie (2001) only provide lower bounds on the competitive ratio of their algorithms in the case of deterministic algorithms, whereas we provide a bound that hold for randomized algorithms. Finally, we provide the optimal structure of policies when arrivals occur according to a Poisson process, which encompasses a rich class of arrival distributions.

1.3 | Notation

We define $\mathcal{S}$ as a finite discrete metric space with distance function $\|\cdot\|$. The depot is located at $s^* \in \mathcal{S}$. An *order* (t, s) is a request for a delivery to location $s \in \mathcal{S}$ placed at time $t \in \mathbb{R}_+$. An *order sequence* of length n is a list of n orders ordered by non-decreasing order arrival time. We define $\sigma^{(n)}$ as a random variable over order sequences of length n with distribution function $\mu^{(n)}$; distribution functions are defined within the context of each section. For example in section 2 we allow $\mu^{(n)}$ to be chosen by an adversary, while in section 3 we require that $\mu^{(n)}$ describe orders that occur according to a Poisson process. For an order sequence $((t_1, s_1), (t_2, s_2), \dots, (t_n, s_n))$ and a departure schedule that delivers order (t_i, s_i) at time t_1', the *latency* of the order is $w_i = t_1' - t_i$. The objective is to minimize $\sum_{i=1}^{n} w_i$. Finally, for any positive integer n we define $[n] \equiv \{1, 2, \dots, n\}$.

2 | ADVERSARIAL ARRIVALS

Definition 1 *Let a randomized online algorithm ALG be given for a minimization problem along with the optimal offline algorithm OPT. If*

$$\mathbb{E}[\mathrm{ALG}(\omega)] \leq c \cdot \mathrm{OPT}(\omega)$$

for all finite order sequences ω chosen in full before ALG is run, then ALG is a randomized c-competitive algorithm against an oblivious adversary. If this holds when ω can be chosen over time as the algorithm responds, then ALG is a randomized c-competitive algorithm against an adaptive online adversary. If this holds when ω can be chosen knowing all random choices the algorithm will make in advance, then ALG is a randomized c-competitive algorithm against an adaptive offline adversary.

We consider the case where the order sequences are chosen by an adversary. We consider both the *oblivious adversary* and the *adaptive offline adversary*. The oblivious adversary must choose the order sequence prior to the online algorithm responding to any of the input; this adversary is thus oblivious to the random choices that the online algorithm will take. The adaptive offline adversary knows the outcomes of all random choices the online algorithm will make before choosing the order sequence; this adversary can thus adapt to any randomization by the online algorithm before the online algorithm takes any action. Since the adaptive offline adversary can behave identically to the oblivious adversary, the competitive ratio of any algorithm against an oblivious adversary is a lower-bound on the competitive ratio of that algorithm against the adaptive offline adversary. We prove that algorithm 1 is $(3\beta\Delta/2\delta - 1)$-competitive against the oblivious adversary

Algorithm 1 RAND-SINGLE

Draw $\alpha \leftarrow \text{Uniform}(0, \beta\Delta)$
for $i \leftarrow 0, 1, \ldots$ **do**
Let ω be the order sequence representing orders waiting to be served at the depot at time $i\beta\Delta + \alpha$
Depart at time $i\beta\Delta + \alpha$ with all orders in ω, visiting customers in the same order as in $\text{TSP}_\beta(\mathcal{S})$ but skipping locations with no delivery
end for

and is $(2\beta\Delta/\delta - 1)$-competitive against the adaptive offline adversary.

Let $\text{TSP}_\beta(X)$ be the length of a traveling salesman tour over X that is within a factor of $\beta \geq 1$ of optimality. Define

$$\Delta = \text{TSP}_1(\mathcal{S}), \tag{1}$$

the length of the optimal traveling salesman tour over $\mathcal{S}$, along with

$$\delta = \min\{\|s - s^*\| | s \in \mathcal{S} \setminus \{s^*\}\}, \tag{2}$$

the minimum distance to a customer from the depot. Our analysis will depend on a polynomial time traveling salesman approximation algorithm TSP_β with approximation guarantee β, where by convention we always include the depot s^* as the starting location of the tour. In particular, algorithm 1 will always visit cities in the same order as $\text{TSP}_\beta(\mathcal{S})$, but will shortcut the tour by skipping over cities that do not have any deliveries.

Throughout this section we define $\text{OPT}(\omega)$ as the optimal cost of serving the order sequence ω in an offline fashion, while $\text{ALG}(\omega)$ is the cost of serving the order sequence ω in an online fashion according to some context-specific algorithm. We show that algorithm 1 is $(3\beta\Delta/2\delta - 1)$-competitive against the oblivious adversary and is $(2\beta\Delta/\delta - 1)$-competitive against the adaptive offline adversary in this setting, and that there do not exist algorithms with a competitive ratio less than $1 + 0.271\Delta/\delta$. Our approach is as follows. We first provide a lower-bound on the cost of the offline optimal algorithm for this problem. Using this bound, we then show that algorithm 1 has the desired competitive ratios against each adversary. We provide instances demonstrating that this analysis is tight. To compute a lower-bound on the competitive ratio of any randomized algorithm we utilize Yao's principle (Borodin & El-Yaniv, 1998) and a carefully chosen sequence of distributions over order sequences.

2.1 | A competitive randomized algorithm

Consider algorithm 1. To motivate this algorithm, consider the options available to the courier when it departs with a number of orders to deliver. The courier must trade off serving the existing orders with minimum latency and returning to the depot as quickly as possible to serve future orders. If the courier decides to minimize the total latency of existing orders then a TRP tour should be used to serve the current orders; if the courier instead decides to return to the depot as quickly as possible then a TSP tour should be used. The length of an optimal TRP tour can be arbitrarily long, since by placing multiple orders to the same location we can force the courier to travel the same path multiple times. For this reason we choose to minimize the time to return to depot to prevent the adversary from punishing us while we are on a long tour. Since we can bound the time we are away from the depot, we can bound the latency of any individual order.

We show that this algorithm has a competitive ratio of $3\beta\Delta/2\delta - 1$. We begin by providing a lower bound on the cost of the offline optimal algorithm in this setting.

Lemma 1 *Let an order sequence $\omega = \{(t_1, s_1), (t_2, s_2), \ldots, (t_n, s_n)\}$ be given. Then $OPT(\omega) \geq n\delta$.*

Proof Consider any order (t_i, s_i) in ω. The courier must depart from s^* and later arrive at s_i to deliver the order. By the triangle inequality, the courier travels at least $\|s_i - s^*\|$ distance to do so. From Equation 2, we have that $\|s_i - s^*\| \geq \delta$. Thus the optimal offline algorithm incurs a cost of at least δ for each order, and so pays at least $n\delta$ in total. ∎

We can use this result to bound the competitive ratio of algorithm 1 in the case of an oblivious adversary.

Theorem 1 *Algorithm 1 is $(3\beta\Delta/2\delta - 1)$-competitive against an oblivious adversary, where $\beta \geq 1$ is the approximation ratio of the TSP, Δ is the length of the optimal Traveling Salesman tour over $\mathcal{S}$, and δ is the minimum distance between the depot s^* and any other location $s \in \mathcal{S} \setminus \{s^*\}$.*

Proof Let an order sequence $\omega = \{(t_1, s_1), (t_2, s_2), \ldots, (t_n, s_n)\}$ be given. We first show that algorithm 1 produces a feasible schedule that serves all orders in ω, that is, a schedule in which the courier always returns to the depot to collect orders before delivering them to their destination. Consider a departure at time $k\beta\Delta + \alpha$ for any $k \geq 0$ and any realization of α, and let $\omega' \subseteq \omega$ be the order sequence representing orders that have not yet been served

at that time. The algorithm departs at $k\beta\Delta + \alpha$ and embarks on a tour according to $\mathrm{TSP}_\beta(S)$, where cities with no order in ω' are skipped. By construction we have that $\mathrm{TSP}_\beta(S) \le \beta\Delta$, and so the courier will return to the depot before the next scheduled departure at $(k+1)\beta\Delta + \alpha$ after having served all orders in ω'.

We now show that the competitive ratio is as claimed. We proceed by bounding the cost of any single order. Let an order (t, s) be given, and define $k = \lfloor t/\Delta \rfloor$. Our algorithm will depart at times $k\Delta + \alpha$ and $(k+1)\Delta + \alpha$. Thus (t, s) will be sent for delivery at time $k\Delta + \alpha$ when $t \le k\Delta + \alpha$ and at time $(k+1)\Delta + \alpha$ when $t > k\Delta + \alpha$. If $t \le k\beta\Delta + \alpha$ the order will wait $k\beta\Delta + \alpha - t$ before departing, and otherwise will wait $(k+1)\beta\Delta + \alpha - t$ before departing.

When the courier departs with this order it will be delivered along a tour of length at most $\mathrm{TSP}_\beta(S) \le \beta\Delta$. Since the tour must begin and end at s^*, we will visit s after traveling at most $\beta\Delta - \|s - s^*\|$. The latency of this order, w_i, consists of the waiting time before the courier departs and the delivery time after it departs. This gives us

$$\begin{aligned} w_i &\le \beta\Delta - \|s - s^*\| + (k\beta\Delta + \alpha - t)1_{\{t \le k\beta\Delta + \alpha\}} \\ &\quad + ((k+1)\beta\Delta + \alpha - t)1_{\{t > k\beta\Delta + \alpha\}} \\ &\le \beta\Delta - \|s - s^*\| + k\beta\Delta + \alpha - t + \beta\Delta 1_{\{\alpha < t - k\beta\Delta\}}. \end{aligned}$$

Since α is uniformly distributed over $[0, \beta\Delta]$, we can compute

$$\begin{aligned} \mathbb{E}[w_i] &\le \beta\Delta - \|s - s^*\| + k\beta\Delta + \frac{\beta\Delta}{2} \\ &\quad - t + \beta\Delta\mathbb{P}[\alpha < t - k\beta\Delta] \\ &\le \beta\Delta - \|s - s^*\| + k\beta\Delta + \frac{\beta\Delta}{2} - t + t - k\beta\Delta \\ &\le \frac{3}{2}\beta\Delta - \|s - s^*\|. \end{aligned}$$

Thus the expected cost of the order sequence ω is

$$\mathbb{E}[\mathrm{ALG}(\omega)] = \mathbb{E}\left[\sum_{i=1}^n w_i\right] = \sum_{i=1}^n \mathbb{E}[w_i] \le \frac{3}{2}n\beta\Delta - \sum_{i=1}^n \|s - s^*\|$$

by the linearity of expectations. Finally, by Lemma 1 we have that

$$\begin{aligned} \mathrm{ALG}(\omega) &\le \frac{3}{2}n\beta\Delta - \sum_{i=1}^n \|s - s^*\| \\ &\le \left(\frac{3\beta\Delta}{2\delta} - 1\right)\mathrm{OPT}(\omega), \end{aligned}$$

as required. ∎

It is worth noting where this proof depends on the assumption of an oblivious adversary. In particular, we use this assumption when we take an expectation over α. An adaptive online adversary can learn α by sending just a single order at time 0 and observing the algorithm's response. An adaptive offline adversary simply knows the realization of α in advance. This leads to the following result.

Theorem 2 *Algorithm 1 is $(2\beta\Delta/\delta - 1)$-competitive against an adaptive offline adversary, and hence against an adaptive online adversary.*

Proof Let an order sequence $\omega = \{(t_1, s_2), (t_2, s_2), \ldots, (t_n, s_n)\}$ be given. From the definition of algorithm 1 there is a departure within $\beta\Delta$ of the arrival of any order. Once on the courier, an order (t_i, s_i) waits at most an additional $\beta\Delta - \|s_i - s^*\|$ time before being delivered, as in the proof of Theorem 1. Thus the latency of the order (t_i, s_i) is at most

$$w_i = \beta\Delta + \beta\Delta - \|s_i - s^*\|,$$

and so we have

$$\mathrm{ALG}(\omega) \le \sum_{i=1}^n (2\beta\Delta - \|s_i - s^*\|) \le \left(\frac{2\beta\Delta}{\delta} - 1\right)\mathrm{OPT}(\omega),$$

as required. ∎

Finally, we note that our analysis of algorithm 1 is tight.

Theorem 3 *There exists an input to algorithm 1 that achieves the $(3\beta\Delta/2\delta - 1)$-competitive ratio against an oblivious adversary. There exist inputs to algorithm 1 that have a competitive ratio arbitrarily close to $(2\beta\Delta/\delta - 1)$ against an adaptive offline adversary.*

Proof We construct an input that achieves the desired results. We use the same metric space $S = \{0, \delta\}$ for any $\delta > 0$, two points on the real line, for both the oblivious and adaptive offline adversaries. Note that δ is also the minimum distance as defined in Equation 2, and so $\Delta = 2\delta$. Here 0 is the depot and δ is the only customer location. In this simple metric space our Traveling Salesman approximation ratio is $\beta = 1$, as the optimal tour is trivially the sequence $0, \delta, 0$.

The order sequence for the oblivious adversary is $\omega = \{(0, \delta)\}$, a single order arriving at time zero to be delivered to the customer at δ. Here $\mathrm{OPT}(\omega) = \delta$, achieved by immediately departing with the order at time zero the customer. The online algorithm waits $\Delta/2$ time in expectation before departing, and so $\mathrm{ALG}(\omega) = \Delta/2 + \delta = 2\delta$. The competitive ratio in this case is then

$$\frac{\mathrm{ALG}(\omega)}{\mathrm{OPT}(\omega)} = \frac{2\delta}{\delta} = 2 = \frac{3\beta\Delta}{2\delta} - 1,$$

as required.

For the adaptive offline adversary we construct a sequence of inputs, and show that in the limit the competitive ratio is tight. Since we are considering an adaptive offline adversary, we can construct input sequences knowing the realization of the random waiting time α from algorithm 1 in advance. For each $n \geq 1$, the order sequence ω_n consists of an order arriving at time zero to be delivered to customer at δ, and $n \geq 1$ orders arriving at time $\alpha + 1/n$ to be delivered to customer at δ. Consider an offline algorithm ALG′ that waits until time $\alpha + 1/n$ to depart with all orders. Then $\text{ALG}'(\omega_n) = (n+1)\delta + \alpha + 1/n$. The online algorithm delivers the first order with latency $\alpha + \delta$, and the remaining orders with latency $3\delta - \frac{1}{n}$ each, giving us $\text{ALG}(\omega_n) = \alpha + \delta + 3n\delta - 1$. Since $\text{OPT}(\omega_n) \leq \text{ALG}'(\omega_n)$, we have that

$$\frac{\text{ALG}(\omega_n)}{\text{OPT}(\omega_n)} \geq \frac{\text{ALG}(\omega_n)}{\text{ALG}'(\omega_n)} = \frac{\alpha + \delta + 3n\delta - 1}{(n+1)\delta + \alpha + 1/n}.$$

Note that α is independent of n. Then

$$\lim_{n\to\infty} \frac{\text{ALG}(\omega_n)}{\text{ALG}'(\omega_n)} \to 3 = \frac{2\beta\Delta}{\delta} - 1,$$

as required. ∎

2.2 | A lower bound on the competitive ratio

We now provide a lower bound on the competitive ratio of any online algorithm by utilizing Yao's Lemma, shown below. We proceed as follows. We first describe an input distribution for inputs of any given length N. We then provide an upper bound on the expected cost of the optimal offline algorithm for this input distribution. Next we show that the optimal deterministic algorithm for any given input will only choose to depart at certain times, and then we will provide a lower bound on the cost of such an algorithm. Finally we will apply these results to Yao's Lemma as we increase $N \to \infty$.

Theorem 4 *(Theorem 8.6 of (5), Yao's principle) Here we consider request-answer systems for which there is no a priori bound on the number of requests, and so the cost of the game may be unbounded. Let ALG be any randomized online algorithm for the request-answer system. Let $\sigma^{(n)}$ be a random variable representing n-length arrival sequences with distribution $\mu^{(n)}$ for all $n \geq 1$. Let $\{ALG_{n,i}\}$ be the set of all deterministic algorithms for serving order sequences of length n. If*

$$\liminf_{n\to\infty} \frac{\inf_i \mathbb{E}_{\mu^{(n)}}[\text{ALG}_{n,i}(\sigma^{(n)})]}{\mathbb{E}_{\mu^{(n)}}[\text{OPT}(\sigma^{(n)})]} \geq c,$$

and

$$\limsup_{n\to\infty} \mathbb{E}_{\mu^{(n)}}[\text{OPT}(\sigma^{(n)})] = \infty,$$

then the competitive ratio of the algorithm is at least c.

We begin by constructing an input distribution $\mu^{(N)}$ over N-length order sequences. To construct $\mu^{(N)}$, let $\overline{S} = S \setminus \{s^*\}$ and let

$$\underline{s} = \arg\min\{\|s - s^*\| | s \in \overline{S}\};$$

that is, $\overline{S}$ represents the full set of customer locations in S while $\underline{s}$ is a location as close to the depot as possible. Define as well

$$m_N = \max\{i \in \mathbb{Z}_+ | (i+1)^2 \leq N\},$$

and so $(m_N + 1)^2 \leq N < (m_N + 2)^2$. Let X be a random variable with mass function $f(i) = 1/m_N$, $i \in [m_N]$, along with *i.i.d.* exponential random variables Y_i with rate parameter λ for each $i \in [X+1]$. Define

$$T_i = \begin{cases} 0, & i = 1, \\ T_{i-1} + Y_i + \Delta + 2\delta, & 2 \leq i \leq X, \\ T_X + Y_{X+1}, & i = X+1. \end{cases}$$

for $i \in [X+1]$.

Our input distribution $\mu^{(N)}$ consists of $Z+1$ *bunches* of arrivals, where a bunch is a collection of orders arriving at the same time. We only consider N such that $m_N > |\overline{S}|$. The bunches arrive at *bunch times* $T_1, T_2, \ldots, T_{Z+1}$. For $i \in [Z]$, the bunch arriving at T_i consists of one order going to each location in $\overline{S}$, along with $m_N - |\overline{S}|$ orders going to $\underline{s}$, for a total of m_N orders. The bunch arriving at time T_{Z+1} consists of $N - m_N Z$ orders all going to $\underline{s}$.

Lemma 2 *Let N be given. Assuming $Z \geq i$, there exists a delivery strategy that incurs total latency of at most*

$$m_N\delta + |\overline{S}|\Delta.$$

for the bunch arriving at time T_i, and that takes at most $\Delta + 2\delta$ time to return to the depot after departing with the orders arriving at time T_i.

Proof We define a sequence of cities to visit that achieves the desired latency and delivery time. Suppose that $\text{TSP}_1(\overline{S})$ gives a tour $s^*, s_1, \ldots, s_k, s^*$, which by construction has length at most Δ. Consider the path $s^*, \underline{s}, s_1, \ldots, s_k$, which may visit $\underline{s}$ twice. From the triangle inequality we have that

$$\|\underline{s} - s_1\| \leq \|\underline{s} - s^*\| + \|s^* - s_1\|,$$

and so this path is no longer than the path $s^*, \underline{s}, s^*, s_1, \ldots, s_k$. Since $\|s^* - \underline{s}\| \leq \delta$, this path has length at most $2\delta + \Delta$. Note that since s_k is at least δ from s^*, all deliveries to locations in $\overline{S}$ travel no more than $\delta + \Delta$. The cost of serving the orders to $\underline{s}$ is $(m_N - |S|)\delta$, while the cost of

serving the remaining orders to locations in $\overline{S}$ is at most $|\overline{S}|(\Delta+\delta)$. Thus the total latency of the orders in the bunch is at most

$$(m_N - |S|)\delta + |\overline{S}|(\Delta+\delta) = m_N\delta + |\overline{S}|\Delta,$$

as required. ∎

Lemma 3 *Let N be given. Assuming $Z \geq i$, the total latency of orders in the bunch arriving at time T_i is at least $m_N\delta$, and the delivery takes at least Δ time.*

Proof By construction $\|s - s^*\| \geq \delta$ for all $s \in \overline{S}$, and so each of the m_N orders in the bunch incurs a cost of at least δ. Finally, by assumption we have that $\mathrm{TSP}_1(\overline{S}) = \Delta$, and so the courier can return from delivery no sooner than Δ after departing. ∎

We now provide an upper bound on the expected cost of the optimal offline algorithm for this input distribution.

Lemma 4 *For any N such that $m_N > |\overline{S}|$,*

$$\mathbb{E}_{\mu^{(N)}}[\mathrm{OPT}(\sigma^{(N)})] \leq \delta N + o(N).$$

Proof We provide an offline algorithm that achieves the desired cost; the optimal offline algorithm must do at least as well. For a given realization of Z, we choose to depart at times T_1, $\ldots$, T_{Z-1}, and then at time T_{Z+1}. Each time we depart we follow the tour described in Lemma 2. Note that since each departure returns us to the depot in at most $\Delta + 2\delta$ time that this departure schedule is feasible. This incurs a cost of at most $m_N\delta + |\overline{S}|\Delta$ for the bunches at times T_1, $\ldots$, T_{Z-1}. Orders in the bunch at time T_Z will wait an additional Y_{Z+1} time before being delivered alongside the orders in the final bunch. When we make the final delivery we follow the same path as in previous bunches, yielding a total cost of $m_N Y_{Z+1} + m_N\delta + |\overline{S}|\Delta$ for orders in the bunch at time T_Z and a total cost of $(N - m_N Z)\delta$ for orders in the final bunch. Thus we have that

$$\begin{aligned}\mathrm{OPT}(\sigma^{(N)}) &\leq (m_N\delta + |\overline{S}|\Delta)Z + m_N Y_{Z+1} + (N - m_N Z)\delta \\ &= N\delta + m_N Y_{Z+1} + Z|\overline{S}|\Delta.\end{aligned}$$

Recall that our metric space $\mathcal{S}$ is finite and independent of N, and so $\overline{S}$ is finite and independent of N as well. Likewise, Δ and δ are functions of $\mathcal{S}$, and so are independent of N. Taking expectations, we find

$$\begin{aligned}\mathbb{E}_{\mu^{(N)}}[\mathrm{OPT}(\sigma^{(N)})] &\leq N\delta + \frac{m_N}{\lambda} + \frac{m_N+1}{2}|\overline{S}|\Delta \\ &\leq N\delta + o(N),\end{aligned}$$

since $\lim_{N\to\infty} m_N/N \to 0$ by construction. ∎

We must now provide lower bounds on the cost of the best deterministic algorithm for any input sequence $\sigma^{[N]}$. We first show that we can restrict our attention to algorithms that depart only at times that are a subset of bunch times $\{T_1, \ldots, T_{Z+1}\}$.

Lemma 5 *For any order sequence $\sigma^{(N)}$ drawn from $\mu^{(N)}$, let any algorithm be given that chooses to depart at some time not in the set $\{T_1, \ldots, T_{Z+1}\}$ when able to do so. Then this algorithm performs no better than an algorithm that chooses only to depart at times in the set $\{T_1, \ldots, T_{Z+1}\}$ when able to do so.*

Proof Let ALG be a deterministic algorithm that chooses to depart at some time not in $\{T_1, \ldots, T_{Z+1}\}$. Since ALG is deterministic, for it to depart at some time not in the set $\{T_1, \ldots, T_{Z+1}\}$ it must choose to depart a fixed time $x > 0$ after some bunch time T_j unless the next order arrives before that time, in which case it can choose to depart at time T_{j+1} or continue waiting. Let T_j be the first time the algorithm chooses to wait $x > 0$ before departing. Note that if $j = X + 1$, then the algorithm is waiting to depart after the final arrival and so is trivially worse than an otherwise equivalent algorithm that chooses $x = 0$. For $j < X + 1$ there exist algorithms that are otherwise equivalent to this one, except that they either choose $x = 0$ or $x = \infty$. Choosing $x = 0$ is the same as choosing to depart at time T_j when possible, whereas choosing $x = \infty$ is the same as choosing not to depart at time T_j and instead departing no sooner than T_{j+1} where possible. Note that choosing $x = T_{j+1} - T_j$ is not possible, as T_{j+1} is unknown an ALG is a deterministic algorithm. Let ALG_1 be the algorithm that chooses $x = 0$ and let ALG_2 be the algorithm that chooses $x = \infty$.

Suppose that $T_j + x < T_{j+1}$, and so ALG departs before the next bunch time. In this case we have that $\mathrm{ALG}(\sigma^{[N]}) \geq \mathrm{ALG}_1(\sigma^{(N)}) + m_N x$, since the m_N orders that arrived at T_j wait an additional x before departure relative to what they wait under ALG_1. Alternatively, if $T_j + x \geq T_{j+1}$ then ALG behaves exactly like ALG_2, and so $\mathrm{ALG}(\sigma^{(N)}) \geq \mathrm{ALG}_2(\sigma^{(N)})$. Let $p = \mathbb{P}[T_j + x < T_{j+1}]$. Then

$$\mathrm{ALG}(\sigma^{(N)}) \geq p\mathrm{ALG}_1(\sigma^{(N)}) + pm_N x + (1-p)\mathrm{ALG}_2(\sigma^{(N)}).$$

Thus either $\mathrm{ALG}_1(\sigma^{(N)}) \leq \mathrm{ALG}(\sigma^{(N)})$ or $\mathrm{ALG}_2 \leq \mathrm{ALG}\,(\sigma^{(N)})$, as required. ∎

We now provide a lower bound on the cost of any deterministic algorithm for a particular choice of λ. Define the function $f(x) = xe^x$, and let $LambertW(x)$ be the inverse of $f(x) = xe^x$.

Lemma 6 *Let ALG be any deterministic algorithm. For*

$$\lambda = \frac{2 + LambertW(-e^{-2})}{\Delta}$$

and $\sqrt{N} > |\overline{S}|$,

$$\mathbb{E}[\mathrm{ALG}(\sigma^N)] \geq m_N(m_N+1)\left(\delta + \frac{1}{2}\cdot\frac{\Delta}{LambertW(-e^{-2})+2}\right).$$

Proof Suppose we are at time T_k, and so the algorithm has just observed the kth bunch. We compute lower bounds on the expected cost of choosing to depart immediately at T_k and the cost of choosing to remain until at least T_{k+1}. Note that we assign the cost of delivering the final bunch to the bunch at time T_Z.

We first consider the cost of departing immediately. By Lemma 3 the algorithm must pay at least $m_N\delta$ to deliver the orders in bunch at time T_k, and the algorithm takes at least Δ time to return to the depot. Additionally, if $Z = k$ there is the chance that the bunch at T_{Z+1} must wait until we return (after no less than Δ time) before being delivered, where again each delivery takes at least δ time. Thus the cost of departing is at least

$$C_{\mathrm{Depart}} = m_N\delta + 1_{\{Z=k|Z>k-1\}}(N - m_N k)(\delta + (\Delta - Y_{Z+1})^+).$$

■

Note that the expressions $1_{\{Z=k|Z>k-1\}}$ and $(\Delta - Y_{Z+1})^+$ are independent, since even though the random variable Z is present in both, in the latter expression it is only an index and each of the Y_i, $i = 1, 2, \ldots$ variables are *i.i.d.* exponential random variables independent of Z. Taking the expectation over Z and Y_{Z+1}, we find

$$\begin{aligned}\mathbb{E}[C_{\mathrm{Depart}}] &= m_N\delta + \frac{N - m_N k}{m_N - k + 1}\left(\delta + \int_0^\Delta (\Delta - y)\lambda e^{-\lambda y}\, dy\right)\\ &= m_N\delta + \frac{N - m_N k}{m_N - k + 1}\left(\delta + \frac{e^{-\lambda\Delta} - 1}{\lambda} + \Delta\right).\end{aligned}$$

Since $N \geq (m_N+1)^2 \geq m_N(m_N+1)$, we have that

$$\begin{aligned}\mathbb{E}[C_{\mathrm{Depart}}] &\geq m_N\delta + m_N\left(\delta + \frac{e^{-\lambda\Delta} - 1}{\lambda} + \Delta\right)\\ &= 2m_N\delta + m_N\left(\frac{e^{-\lambda\Delta} - 1}{\lambda} + \Delta\right).\end{aligned}$$

We now consider the cost of choosing to remain at the depot until at least time T_{k+1}. Since $T_{k+1} - T_k = Y_{k+1} + \Delta + 2\delta \geq Y_{k+1}$, the total cost of delivering these orders is at least $m_N Y_{k+1} + m_N\delta$. If $Z = k$ we must also pay for the orders in the last bunch. This gives us that the cost of remaining is at least

$$C_{\mathrm{Remain}} = m_N Y_{k+1} + m_N\delta + 1_{\{Z=k|Z>k-1\}}(N - m_N k)\delta.$$

Taking expectations, we find

$$\begin{aligned}\mathbb{E}[C_{\mathrm{Remain}}] &= \frac{m_N}{\lambda} + m_N\delta + \frac{N - m_N k}{m_N - k + 1}\delta\\ &\geq \frac{m_N}{\lambda} + 2m_N\delta.\end{aligned}$$

To bound the cost of the algorithm it will be sufficient to choose λ and $\varphi > 0$ such that

$$\min\{\mathbb{E}[C_{\mathrm{Depart}}], \mathbb{E}[C_{\mathrm{Remain}}]\} \geq \phi;$$

if this holds, then we incur at least ϕ at each bunch time. Consider

$$\lambda = \frac{2 + \mathrm{LambertW}(-e^{-2})}{\Delta}.$$

Note that this implies that

$$\begin{aligned}e^{-\lambda\Delta} &= e^{-2-\mathrm{LambertW}(-e^{-2})}\\ &= e^{-2}e^{-\mathrm{LambertW}(-e^{-2})}\\ &= e^{-2}\frac{\mathrm{LambertW}(-e^{-2})}{-e^{-2}}\\ &= -\mathrm{LambertW}(-e^{-2}).\end{aligned}$$

Then

$$\begin{aligned}\mathbb{E}[C_{\mathrm{Depart}}] &\geq 2m_N\delta + m_N\left(\frac{e^{-\lambda\Delta} - 1}{\lambda} + \Delta\right)\\ &\geq 2m_N\delta + m_N\Delta\left(1 - \frac{\mathrm{LambertW}(-e^{-2}) + 1}{\mathrm{LambertW}(-e^{-2}) + 2}\right)\\ &\geq 2m_N\delta + \frac{m_N\Delta}{\mathrm{LambertW}(-e^{-2}) + 2},\end{aligned}$$

while

$$\begin{aligned}\mathbb{E}[C_{\mathrm{Remain}}] &\geq 2m_N\delta + \frac{m_N}{\lambda}\\ &\geq 2m_N\delta + \frac{m_N\Delta}{\mathrm{LambertW}(-e^{-2}) + 2}.\end{aligned}$$

From Lemma 5 we know that we can restrict our attention to algorithms which only choose to depart at bunch times. Any such algorithm incurs a cost of at least

$$2m_N\delta + \frac{m_N\Delta}{\mathrm{LambertW}(-e^{-2}) + 2}$$

at each of the first Z bunch times. This gives us

$$\begin{aligned}&\mathbb{E}_{\mu^{(N)}}[\mathrm{ALG}(\sigma^{(N)})]\\ &\geq \mathbb{E}\left[\sum_{i=1}^{Z}\left(2m_N\delta + \frac{m_N\Delta}{\mathrm{LambertW}(-e^{-2}) + 2}\right)\right]\\ &= \mathbb{E}[Z]\left(2m_N\delta + \frac{m_N\Delta}{\mathrm{LambertW}(-e^{-2}) + 2}\right)\\ &= \frac{m_N + 1}{2}\left(2m_N\delta + \frac{m_N\Delta}{\mathrm{LambertW}(-e^{-2}) + 2}\right)\\ &= m_N(m_N+1)\left(\delta + \frac{1}{2}\cdot\frac{\Delta}{\mathrm{LambertW}(-e^{-2}) + 2}\right).\end{aligned}$$

We are now prepared to provide a lower bound on the competitive ratio of any online algorithm via Yao's principle.

Theorem 5 *There does not exist an online algorithm with competitive ratio less than* $1 + 0.271\Delta/\delta$.

Proof We apply Yao's principle to our input distribution $\mu^{(N)}$ with

$$\lambda = \frac{2 + \text{LambertW}(-e^{-2})}{\Delta}.$$

To begin, note that the second assumption of Yao's principle is satisfied due to Lemma 1, which ensures that the cost incurred by any algorithm on an order sequence of length n grows at least linearly in n. We now show that the first assumption of Yao's principle holds. From Lemma 4 we have that

$$\begin{aligned}\mathbb{E}_{\mu^{(N)}}[\text{OPT}(\sigma^{(N)})] &\leq \delta N + \left(1 + \frac{1}{\lambda}\right) o(N) \\ &\leq \delta N + o(N)\end{aligned}$$

since λ does not depend on N. Likewise, from Lemma 6 we have that

$$\begin{aligned}&\inf_i \mathbb{E}_{\mu^{(N)}}[\text{ALG}_{N,i}] \\ &\geq m_N(m_N+1)\left(\delta + \frac{1}{2}\cdot\frac{\Delta}{\text{LambertW}(-e^{-2})+2}\right) \\ &\geq m_N^2\left(\delta + \frac{1}{2}\cdot\frac{\Delta}{\text{LambertW}(-e^{-2})+2}\right) + o(N),\end{aligned}$$

where $\{\text{ALG}_{N,i} | i \in \mathbb{Z}_+\}$ is the set of all deterministic algorithms for order sequences of length N. Thus

$$\begin{aligned}&\lim_{N\to\infty} \frac{\inf_i \mathbb{E}_{\mu^{(N)}}[\text{ALG}_{N,i}]}{\mathbb{E}_{\mu^{(N)}}[\text{OPT}(\sigma^{(N)})]} \\ &\geq \lim_{N\to\infty} \frac{m_N^2\left(\delta + \frac{1}{2}\cdot\frac{\Delta}{\text{LambertW}(-e^{-2})+2}\right) + o(N)}{\delta N + \left(1+\frac{1}{\lambda}\right)o(N)} \\ &= 1 + \frac{\frac{1}{2}\cdot\frac{\Delta}{\text{LambertW}(-e^{-2})+2}}{\delta} \\ &= 1 + \frac{1}{2(\text{LambertW}(-e^{-2})+2)}\cdot\frac{\Delta}{\delta} \\ &> 1 + 0.271\frac{\Delta}{\delta},\end{aligned}$$

as required. ∎

The bound of Theorem 5 was constructed for oblivious adversaries. Since adaptive adversaries can choose any input that an oblivious adversary can, this bound holds for adaptive adversaries as well.

3 | AVERAGE-CASE SETTING

We now consider the case where orders occur according to a Poisson process. We consider order sequences of unbounded length, and seek to minimize the long-run average cost of serving such order sequences. The times of orders will be distributed according to a Poisson process with rate λ, and the location of the orders will be distributed *i.i.d.* according to some probability mass function $f_S : \mathcal{S} \to \mathbb{R}_+$; this ensures that each order sequence $(t_1, s_1), (t_2, s_2), \ldots$ is distributed according to a marked Poisson process. We assume without loss of generality that $f_S(s) > 0$ for all $s \in \mathcal{S} \setminus \{s^*\}$.

In Section 2 we considered online algorithms that made use of approximation algorithms to produce an a priori TSP tour. Here we only rely on having any a priori TSP tour Π over all locations in $\mathcal{S}$. We make no assumptions about the quality of this tour; rather, for any such tour we derive the structure of the optimal policy for serving orders using that tour. For the remainder of this section we will assume that some Π is given and fixed, and without loss of generality we assume that Π visits $s_1, s_2, \ldots, s_n, s^*$ in that order.

We will further relax our assumption of travel times. In particular we assume that any path through $\mathcal{S}$ of length d will take d time to travel in expectation, with the actual time required to traverse the path being exponentially distributed with mean d. This will make the problem amenable to analysis as a continuous time Markov decision process (CTMDP) and as a Markov decision process (MDP). Deterministic travel times would introduce state transitions that are not exponentially distributed, causing the evolution of the system to depend on its history rather than only its current state. To avoid this dependency on history we would need to greatly increase the size of the state space, making our analysis intractable.

3.1 | The problem as a CTMDP and MDP

We express this problem as CTMDP. We define the state space $X = \mathbb{Z}_+^{|\mathcal{S}|}$. We index elements $x \in X$ by the location each coordinate represents, and so x_s represents the number of orders to $s \in \mathcal{S}$ waiting at the depot. For $x, x' \in X$, we say that $x \leq x'$ if $x_s \leq x_s'$ for all $s \in \mathcal{S}$. For each $s \in [S]$ we define e^s as a vector in X such $e^s_{s'} = 1_{\{s=s'\}}$. For any $x \in X$, define

$$\mathcal{L}(x) = \{s \in \mathcal{S} | x_s > 0, s \neq s^*\} \tag{3}$$

as the set of delivery locations for orders waiting at the depot. For each $x \in X$, let $\Pi(x)$ be the tour that begins at s^*, visits all locations in $\mathcal{L}(x)$ in the same order as in Π, and then returns to s^*.

Define the action Remain as corresponding to remaining at the depot until at least the next arrival. For each $x \in X \setminus \mathbf{0}$, where $\mathbf{0} \in X$ is the zero vector, define the action Depart_x as corresponding to leaving the depot to deliver waiting orders on the tour $\Pi(x)$. We define $\mathcal{A} = \{\text{Remain}\} \cup \{\text{Depart}_x | x \in X\}$ as our action space. The allowed actions in state $x \in X$ are

$$\mathcal{A}_x = \{\text{Remain}\} \cup \{\text{Depart}_{x'} | x' \in X \setminus \mathbf{0}, \mathcal{L}(x) \subseteq \mathcal{L}(x')\}. \tag{4}$$

Note that in state $x \in X \setminus \{\mathbf{0}\}$ we are allowed to depart as if we were in state $\sum_{s\in\mathcal{L}(x)} e^s$, since these actions will result in the same tour. Since the notation $\sum_{s\in\mathcal{L}(x)} e^s$ is cumbersome, we will allow $\mathcal{L}(x)$ to also be defined as $\sum_{s\in\mathcal{L}(x)} e^s$ depending

on context. This allows us to use the less cumbersome notation $\text{Depart}_{\mathcal{L}(x)}$, along with $\Pi(\mathcal{L}(x))$ to indicate a tour going through all delivery locations in x.

The decision epochs will correspond to order arrival times and the times the courier returns to the depot. This is due to the assumption of exponentially-distributed travel times. When the courier departs from the depot, both the time it will take to return to the depot and the time to the next arrival are exponentially distributed, and so the time of the next event—the courier returning to the depot, or the next arrival occurring—is also exponentially distributed.

For each $x \in X$ and $s \in \mathcal{L}(x)$, define $\Pi(x)_s$ as the distance traveled along the tour $\Pi(x)$ before arriving at location s. Note that $\Pi(x)_{s^*}$ indicates the length of the tour, as the final arrival in any tour is at the depot. Choosing the action $\text{Depart}_{x'}$ in state x means that the courier will depart from the depot to deliver all waiting orders, following the tour $\Pi(x')$. At this time we will pay the costs associated with each order being sent along the tour $\Pi(x')$, as well as paying the holding fees associated with any new orders that arrive while we are away from the depot.

When we depart on a tour $\Pi(x')$ in state x, we will be gone for a random time Λ that is exponentially distributed with mean $\Pi(x')_{s^*}$. Since the arrival sequence is a Poisson process with rate λ, given the value of Λ the number of arrivals I is a Poisson random variable with parameter $\lambda\Lambda$. Let $\xi_1, \xi_2, \ldots, \xi_I$ be random variables describing the arrival times of orders while we are away from the depot, measured from the departure time, and let $S_1, S_2, \ldots, S_I$ be the *i.i.d.* random variables describing the locations they are sent to. Conditional on Λ these I arrivals will arrive at times after we depart that are uniformly distributed over $[0, \Lambda]$. We charge each of these arrivals the mean waiting time they accrue, which will be $\Lambda/2$. Thus the total expected waiting time accrued is

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{I}(\Lambda - \xi_i)\right] &= \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^{I}(\Lambda - \xi_i)\middle|\Lambda\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{\Lambda}{2}I\middle|\Lambda\right]\right] \\
&= \frac{\lambda}{2}\mathbb{E}[\Lambda^2] \\
&= \frac{\lambda}{2}\cdot 2\Pi(x')_{s^*}^2 \\
&= \lambda\Pi(x')_{s^*}^2.
\end{aligned}$$

The cost of delivering the orders in x is simply $\sum_{s\in S}\Pi(x')_s x_s$.

As in Sennott (2009), we now need to define transition probabilities P, fixed costs G, rate costs g, and transition rates ν of

$$P_{x,\ x''}(\text{Depart}_{x'}) = \mathbb{P}\left[x'' = \sum_{i\in[I]} e^{S_i}\right], \tag{5}$$

$$G(x, \text{Depart}_{x'}) = \lambda\Pi(x')_{s^*}^2 + \sum_{s\in S}\Pi(x')_s x_s \tag{6}$$

$$g(x, \text{Depart}_{x'}) = 0, \tag{7}$$

$$\nu(x, \text{Depart}_{x'}) = \frac{1}{\Pi(x')_{s^*}}. \tag{8}$$

The transition probabilities P represent that we transition to the random state $\sum_{i\in[I]} e^{S_i}$. The fixed costs G represent the cost of delivering the order currently waiting at the depot, along with the waiting time accrued by new orders that arrive while we are gone. There are no rate costs, since the waiting time accrued by orders on the courier have been accounted for in the fixed costs. Finally, the transition rate ν represents that the delivery takes $\Pi(x')_{s^*}$ in expectation to complete.

Taking the Remain action in state x means that we will not depart from the depot until at least the time of the next order arrival. During this time we pay the holding fees for each order waiting at the depot. The next order will occur in an exponentially distributed time with rate λ and will be sent to a destination in S according to distribution function f_S. While we are waiting for this arrival all orders at the depot continue to accrue waiting time. We have no fixed costs. This gives us transition probabilities P, fixed costs G, rate costs g, and transition rates ν of

$$P_{x,\ x''}(\text{Remain}) = \sum_{s\in S} f_S(s)\cdot 1_{\{x''=x+e^s\}} \tag{9}$$

$$G(x, \text{Remain}) = 0 \tag{10}$$

$$g(x, \text{Remain}) = \sum_{s\in S} x_s, \tag{11}$$

$$\nu(x, \text{Remain}) = \lambda. \tag{12}$$

The transition probabilities P represent that the next state has exactly one additional order. The rate costs g represent the additional waiting time accrued by the waiting orders; since there are $\sum_{s\in S} x_s$ orders waiting at the depot until the next event, we accrue costs at a rate $\sum_{s\in S} x_s$ until the next event. The transition rate ν represents that the next order arrives at an exponential rate with rate parameter λ.

Finally, we can use standard uniformization techniques from Sennott (2009) to convert the CTMDP $\Psi = (S, \mathcal{A}, g, G, \nu, P)$ to a discrete time MDP. The idea behind uniformization is to create a discrete-time MDP where each transition represents a time step

$$\begin{aligned}
\tau &= \inf_{x\in X, a\in\mathcal{A}_x} \nu(x,a)^{-1} \\
&= \min\left\{\frac{1}{\lambda}, \Pi(e^{\underline{s}})_{s^*}\right\}
\end{aligned} \tag{13}$$

the fastest mean transition time in Ψ. This holds because $\Pi(e^{\underline{s}})$ is the shortest tour we can embark on, as there are no locations closer to the depot than $\underline{s}$; thus, the shortest transition time is the smaller of the mean inter-arrival time $1/\lambda$ and the length of the shortest non-empty tour $\Pi(e^{\underline{s}})_{s^*}$. We compensate for slower transitions by increasing the probability of self transitions in those states. The additional self transitions effectively cause us to transition out of these states at the appropriate

lower rate. Applying these techniques gives us a cost function C and transition probabilities P^* defined as

$$C(x,a) = G(x,a)\nu(x,a) + g(x,a) \tag{14}$$

$$P^*_{x,\ x'}(a) = \begin{cases} \tau\nu(x,a)P_{x,\ x'}(a), & x \neq x' \\ 1 - \tau\nu(x,a), & x = x'. \end{cases} \tag{15}$$

It is worth expanding the definition of C, which gives us

$$C(x, \text{Remain}) = \sum_{s\in\mathcal{S}} x_s$$

$$C(x, \text{Depart}_{x'}) = \lambda\Pi(x')_{s^*} + \sum_{s\in\mathcal{S}} \frac{\Pi(x')_s}{\Pi(x')_{s^*}} x_s. \tag{16}$$

With these we can define the MDP $\Phi = (\mathcal{S}, \mathcal{A}, C, P^*)$ as the discrete-time analogue of the CTMDP Ψ.

3.2 | Structural results

Our goal in this section will be to show that average cost optimal policies for Ψ are *threshold policies*, defined as follows.

Definition 2 *A policy $\pi : X \to \mathcal{A}$ is a threshold policy if for any $x \in X$ such that $\pi(x) \neq$ Remain, then for all $x \leq x'$ with $\mathcal{L}(x) = \mathcal{L}(x')$ we have that $\pi(x') \neq$ Remain.*

Note that if π is a threshold policy, $x \leq x'$, $\mathcal{L}(x) \subset \mathcal{L}(x')$, and $\pi(x) \neq$ Remain, then it is not required that $\pi(x') \neq$ Remain.

Lemma 7 *Let x, $x' \in X$ be given with $\mathcal{L}(x) \subseteq \mathcal{L}(x')$. Then $\Pi(x)_s \leq \Pi(x')_s$ for all $s \in \mathcal{L}(x)$.*

Proof We show that $\Pi(x)_s \leq \Pi(x + e^{s'})_s$ for any $s' \in \mathcal{S}$; the result then immediately follows. Let $s \in \mathcal{L}(x)$ and $s' \in \mathcal{S}$ be given. If $s \leq s'$, then $\Pi(x)$ and $\Pi(x+e^{s'})$ follow the same path to s, and so travel the same distance to s. If $s > s'$, there are two cases to consider. First, suppose $x_{s'} > 0$. In this case both $\Pi(x)$ and $\Pi(x + e^{s'})$ follow the same path to s, and so travel the same distance to s. Second, suppose $x_{s'} = 0$. In this case the path to s in $\Pi(x + e^{s'})$ detours from the path given by $\Pi(x)$ to visit s' before reaching s; by the triangle inequality this detour cannot decrease the length of the path to s, and so $\Pi(x)_s \leq \Pi(x + e^{s'})_s$. ∎

Lemma 8 *Let x_1, x_1', x_2, $x_2' \in X$ be given with $\mathcal{L}(x_1) = \mathcal{L}(x_1') = \mathcal{L}(x_2) = \mathcal{L}(x_2')$. Then $P_{x_1,x''}(Depart_{x_1'}) = P_{x_2,x''}(Depart_{x_2'})$ for all $x'' \in X$.*

Proof From Equation 5,

$$P_{x_1,\ x''}(\text{Depart}_{x_1'}) = \mathbb{P}\left[x'' = \sum_{i\in[I]} e^{S_i}\right]$$

$$P_{x_2,\ x''}(\text{Depart}_{x_2'}) = \mathbb{P}\left[x'' = \sum_{i\in[I']} e^{S_i}\right],$$

where I conditional on Λ is a Poisson random variable with rate parameter $\lambda\Lambda$, Λ is an exponentially distributed random variable with rate parameter $\Pi(x_1')_{s^*}$, and I' and Λ' are defined analogously. Since $\mathcal{L}(x_1') = \mathcal{L}(x_2')$, from Lemma 7 we have that $\Pi(x_1')_{s^*} = \Pi(x_2')_{s^*}$, and so I and I' are identically distributed. This ensures that

$$P_{x_1,x''}(\text{Depart}_{x_1'}) = P_{x_2,x''}(\text{Depart}_{x_2'}),$$

as required. ∎

We now show results about the behavior of the discounted value function associated with the MDP Φ. Using the notation of Sennott (2009), for all $\alpha \in (0,1)$, $V_\alpha : X \to \mathbb{R}$ is the α-discounted value function, where $V_\alpha(x)$ represents the optimal expected α-discounted cost when starting in state x. Here α-discounted means that costs in each successive time period are weighted an additional factor of α. Likewise, $V_{\alpha,\pi} : X \to \mathbb{R}$ is the α-discounted value function associated with policy $\pi : X \to \mathcal{A}$, where $V_{\alpha,\pi}(x)$ is the expected discounted cost when starting in state x and following policy π.

Lemma 9 For the MDP Φ and for all $\alpha \in (0, 1)$, $V_\alpha(x) \leq V_\alpha(x')$ for all $x \leq x'$.

Proof We show that for any $s' \in \mathcal{S}$, $V_\alpha(x) \leq V_\alpha(x + e^{s'})$; the result then immediately follows. We argue via coupling. Let π^* be a stationary policy realizing V_α as per Theorem 4.1.4 of Sennott (2009), and let $s' \in \mathcal{S}$ be given. Let T be the first random transition at which π^* does not choose the Remain action given that we begin in state $x + e^s$, noting that it is possible that $T = \infty$ if π^* never does so, and let $\hat{x} + e^{s'}$ be the random state that π^* observes at transition T. Define the non-stationary policy π that chooses the Remain action for the first $T - 1$ transitions, chooses the $\text{Depart}_{\hat{x}+e^{s'}}$ at transition T, and then follows π^* exactly for all remaining transitions. We show that π incurs no more cost than π^* along this sample path. ∎

By construction both π and π^* take the Remain action for the first $T - 1$ transitions. During each of these transitions the cost incurred by π is strictly less than the cost incurred by π^*, since from Equation 16

$$\begin{aligned} C(x, \text{Remain}) &= \sum_{s\in\mathcal{S}} x_s \\ &< \sum_{s\in\mathcal{S}} x_s + 1 \\ &= \sum_{s\in\mathcal{S}} (x + e^{s'})_s \\ &= C(x + e^{s'}, \text{Remain}) \end{aligned}$$

for any $x \in X$. At transition T both π and π^* choose the $\text{Depart}_{\hat{x}+e^{s'}}$ action, and follow the tour $\Pi(\hat{x}+e^{s'})$. Since

$$\begin{aligned} C(\hat{x}, \text{Depart}_{\hat{x}+e^{s'}}) &= \lambda \Pi(\hat{x}+e^{s'})_{s^*} + \sum_{s\in S} \frac{\Pi(\hat{x}+e^{s'})_s}{\Pi(\hat{x}+e^{s'})_{s^*}} \hat{x}_s \\ &\le \lambda \Pi(\hat{x}+e^{s'})_{s^*} + \sum_{s\in S} \frac{\Pi(\hat{x}+e^{s'})_s}{\Pi(\hat{x}+e^{s'})_{s^*}} (\hat{x}_s + e^{s'}) \\ &= C(\hat{x}+e^{s'}, \text{Depart}_{\hat{x}+e^{s'}}), \end{aligned}$$

π again incurs a cost of no more than that incurred by π^*. Both policies will return to the depot in the same state, having departed at the same time and for the same duration. Since from this point onward π follows π^* exactly, both policies incur identical costs moving forward. Thus $V_\alpha(x) \le V_\alpha(x + e^{s'})$, as required.

Theorem 6 *Let $\alpha \in (0, 1)$ be given. There exists a threshold policy that is optimal for the α-discounted MDP Φ.*

Proof Let an optimal policy π^* and $x \in X$ be given. We will show that if $\pi^*(x) \neq \text{Remain}$, then $\pi^*(x + e^s) \neq \text{Remain}$, proving our claim; for the remainder of this proof let $s \in \mathcal{L}(x)$ be given. If $\pi^*(x) = \text{Remain}$, our claim is trivially true. Otherwise, let $\pi^*(x) = \text{Depart}_{x'}$. From Equation 4, it must be that $\mathcal{L}(x) \subseteq \mathcal{L}(x')$, and so the $\text{Depart}_{x'}$ action is a valid choice in state $x + e^s$. From Theorem 4.1.4 of Sennott (2009) we have that

$$V_\alpha(x) = \min\{V_{\alpha,\ \text{Remain}}(x),\ \ V_{\alpha,\ \text{Depart}}(x)\},$$

where

$$\begin{aligned} V_{\alpha,\ \text{Remain}}(x) &= C(x, \text{Remain}) \\ &\quad + \alpha \sum_{x''\in X} P^*_{x,\ x''}(\text{Remain}) V_\alpha(x'') \\ V_{\alpha,\ \text{Depart}}(x) &= C(x, \text{Depart}_{x'}) \\ &\quad + \alpha \sum_{x''\in X} P^*_{x,\ x''}(\text{Depart}_{x'}) V_\alpha(x''), \end{aligned}$$

the expected α-discounted cost of choosing the actions Remain and $\text{Depart}_{x'}$ in state x, respectively. From Equation 16 we have that

$$\begin{aligned} C(x + e^s, \text{Depart}_{x'}) &= \lambda \Pi(x')_{s^*} + \sum_{s'\in S} \frac{\Pi(x')_{s'}}{\Pi(x')_{s^*}} (x + e^s)_{s'} \\ &= \lambda \Pi(x')_{s^*} + \sum_{s'\in S} \frac{\Pi(x')_{s'}}{\Pi(x')_{s^*}} x_{s'} + \frac{\Pi(x')_s}{\Pi(x')_{s^*}} \\ &= C(x, \text{Depart}_{x'}) + \frac{\Pi(x')_s}{\Pi(x')_{s^*}}. \end{aligned}$$

Combining this with Lemma 8, we have that

$$\begin{aligned} V_{\alpha,\ \text{Depart}}(x + e^s) &= C(x + e^s, \text{Depart}_{x'}) \\ &\quad + \alpha \sum_{x''\in X} P^*_{x+e^s,\ x''}(\text{Depart}_{x'}) V_\alpha(x'') \\ &= C(x, \text{Depart}_{x'}) + \frac{\Pi(x')_s}{\Pi(x')_{s^*}} \\ &\quad + \alpha \sum_{x''\in X} P^*_{x,\ x''}(\text{Depart}_{x'}) V_\alpha(x'') \\ &= V_{\alpha,\ \text{Depart}}(x) + \frac{\Pi(x')_s}{\Pi(x')_{s^*}} \\ &< V_{\alpha,\ \text{Depart}}(x) + 1, \end{aligned}$$

since by construction $\Pi(x')_s < \Pi(x')_{s^*}$. Since π^* is an optimal policy and chose to depart in x we have that $V_\alpha(x) = V_{\alpha,\ \text{Depart}}(x) \le V_{\alpha,\ \text{Remain}}(x)$. Then

$$\begin{aligned} V_{\alpha,\ \text{Depart}}(x + e^s) &< V_{\alpha,\ \text{Depart}}(x) + 1 \\ &= V_\alpha(x) + 1 \\ &\le V_{\alpha,\ \text{Remain}}(x) + 1. \end{aligned}$$

From Equation 9 we have that

$$\begin{aligned} P_{x,\ x''}(\text{Remain}) &= \sum_{s'\in S} f_S(s') \cdot 1_{\{x''=x+e^{s'}\}} \\ &= P_{x+e^s,\ x''+e^s}(\text{Remain}). \end{aligned}$$

From Equation 16 we have that

$$\begin{aligned} C(x, \text{Remain}) + 1 &= \sum_{s'\in S} x_{s'} + 1 = \sum_{s'\in S} (x + e^s)_{s'} \\ &= C(x + e^s, \text{Remain}). \end{aligned}$$

Combining this with Lemma 9, we have that $V_{\alpha,\ \text{Remain}}(x) + 1 \le V_{\alpha,\ \text{Remain}}(x + e^s)$, and so

$$V_{\alpha,\ \text{Depart}}(x + e^s) < V_{\alpha,\ \text{Remain}}(x + e^s).$$

Thus $\pi^*(x) \neq \text{Remain}$ implies that $\pi^*(x + e^s) \neq \text{Remain}$ for all $s \in \mathcal{L}(x)$, as required. ∎

Thus we have shown that in the α-discounted setting, optimal policies for Φ exhibit a threshold structure. We now argue that this result holds in the undiscounted case using the following result from Sennott (2009).

Theorem 7 *(Theorem 7.2.3 of Sennott (2009)) Consider the following conditions for the MDP $\Phi = (X, \mathcal{A}, C, P)$, and define $h_\alpha(x) = V_\alpha(x) - V_\alpha(x^*)$ for some distinguished $x^* \in X$.*

SEN 1. *The quantity $(1 - \alpha)V_\alpha(x^*)$ is bounded for all $\alpha \in (0, 1)$.*

SEN 2. *There exists $M \ge 0$ such that $h_\alpha(x) \le M$ for all $x \in X$ and all $\alpha \in (0, 1)$.*

SEN 3. *There exists $L \ge 0$ such that $-L \le h_\alpha(x)$ for all $x \in X$ and all $\alpha \in (0, 1)$.*

Let π^*_α be the α-discount optimal policy for Φ. If Φ satisfies SEN 1—3 and there exists a sequence $\alpha_n \to 1$ such that $\lim_{\alpha_n \to 1} \pi^*_{\alpha_n} = \pi$, then π is average cost optimal.

We must first show that Φ satisfies the conditions of Theorem 7.

Lemma 10 Consider the policy π defined as

$$\pi(x) = Depart_{\mathcal{L}(\overline{S})}. \tag{17}$$

Then $V_{\alpha,\pi}(\mathbf{0}) \leq 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}/(1-\alpha)$.

Proof We begin by providing an upper bound on the cost incurred in each state. From Equation 16, Lemma 7, and the fact that $\Pi(\mathcal{L}(\overline{S}))_s \leq \Pi(\mathcal{L}(\overline{S}))_{s^*}$ for all $s \in \mathcal{S}$, we have that

$$\begin{aligned} C(x, \pi(x)) &= \lambda\Pi(\mathcal{L}(\overline{S}))_{s^*} + \sum_{s\in\mathcal{S}} \frac{\Pi(\mathcal{L}(\overline{S}))_s}{\Pi(\mathcal{L}(\overline{S}))_{s^*}} x_s \\ &\leq \lambda\Pi(\mathcal{L}(\overline{S}))_{s^*} + \sum_{s\in\mathcal{S}} x_s \end{aligned}$$

Thus the cost incurred at each transition is upper bounded by a function that is linear in the number of orders in the state. Applying Lemma 8 gives us that

$$P^*_{x,\ x'}(\text{Depart}_{x+\mathcal{L}(\overline{S})}) = P^*_{\mathcal{L}(\overline{S}),\ x'}(\text{Depart}_{\mathcal{L}(\overline{S})}),$$

for all $x' \in X$, and so at each transition the transition probabilities are independent of the current state. In particular, we transition to the random state $\chi = \sum_{i\in[I]} e^{s_i}$, where I given Λ is a Poisson random variable with rate parameter $\lambda\Lambda$ and Λ is an exponential random variable with mean $\Pi(\mathcal{L}(\overline{S}))_{s^*}$. This gives us

$$\begin{aligned} V_{\alpha,\pi}(\mathbf{0}) &\leq \sum_{t=0}^{\infty} \alpha^t \mathbb{E}[C(\chi, \text{Depart}_{\mathcal{L}(\overline{S})})] \\ &\leq \sum_{t=0}^{\infty} \alpha^t \mathbb{E}\left[\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*} + \sum_{s\in\mathcal{S}} \chi_s\right]. \end{aligned}$$

Straightforward calculation shows that

$$\begin{aligned} \mathbb{E}[I] &= \mathbb{E}[\mathbb{E}[I|\Lambda]] \\ &= \mathbb{E}[\lambda\Lambda] \\ &= \lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}, \end{aligned}$$

and so

$$\begin{aligned} V_{\alpha,\pi}(\mathbf{0}) &\leq 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*} \sum_{t=0}^{\infty} \alpha^t \\ &= \frac{2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}}{1-\alpha}. \end{aligned}$$

■

Lemma 11 *The MDP Φ satisfies SEN 1 to 3 of Theorem 7.*

Proof We show that each of the assumptions holds in turn for Φ. We take the zero vector $\mathbf{0}$ as our distinguished state. Throughout this proof it is assumed that $\alpha \in (0,1)$ is given.

SEN 1. Since all costs C are nonnegative it suffices to show that $(1-\alpha)V_\alpha(\mathbf{0})$ is bounded above. Lemma 10 immediately provides this bound, since any feasible policy provides an upper bound on the cost of the optimal policy.

SEN 2. We show that

$$h_\alpha(x) \leq \frac{2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}}{P^*_{\mathcal{L}(\overline{S}),\ \mathbf{0}}(\text{Depart})},$$

or equivalently that

$$V_\alpha(x) \leq \frac{2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}}{P^*_{\mathcal{L}(\overline{S}),\ \mathbf{0}}(\text{Depart})} + V_\alpha(\mathbf{0}).$$

To do so, we analyze the non-stationary policy $\widehat{\pi}$ that follows π from Equation 17 in states $x \neq \mathbf{0}$ until the first time it reaches state $\mathbf{0}$, and after which it follows some optimal policy π^*.

Let $x \neq \mathbf{0}$ be given, and consider following $\widehat{\pi}$. Note that since

$$\begin{aligned} \tau &= \min\left\{\frac{1}{\lambda}, \Pi(e^{s})_{s^*}\right\} \\ \nu(x, \text{Depart}_{x+\mathcal{L}(\overline{S})}) &= \frac{1}{\Pi(x')_{s^*}}, \\ P_{x,\ \mathbf{0}}(\text{Depart}_{x+\mathcal{L}(\overline{S})}) &= \mathbb{P}\left[\mathbf{0} = \sum_{i\in[I]} e^{S_i}\right], \end{aligned}$$

are nonzero from Equations 13, 8, 5, respectively, we have that

$$\begin{aligned} &P^*_{x,\ \mathbf{0}}(\text{Depart}_{x+\mathcal{L}(\overline{S})}) \\ &= \tau\nu(x, \text{Depart}_{x+\mathcal{L}(\overline{S})}) P_{x,\ \mathbf{0}}(\text{Depart}_{x+\mathcal{L}(\overline{S})}) > 0 \end{aligned}$$

as well. From Lemma 8 we have that

$$\begin{aligned} P^*_{x,\ \mathbf{0}}(\text{Depart}_{x+\mathcal{L}(\overline{S})}) &= P^*_{x+\mathcal{L}(\overline{S}),\ \mathbf{0}}(\text{Depart}_{\mathcal{L}(\overline{S})}) \\ &= P^*_{\mathcal{L}(\overline{S}),\ \mathbf{0}}(\text{Depart}_{\mathcal{L}(\overline{S})}). \end{aligned}$$

This means that at each transition the probability we transition to $\mathbf{0}$ is at least $P^*_{\mathcal{L}(\overline{S}),\ \mathbf{0}}(\text{Depart})$. Thus an upper bound on the number of transitions we need to take until we get to state $\mathbf{0}$ is X, where $X \in \{1, 2, \dots\}$ is a geometric random variable with success probability $P^*_{\mathcal{L}(\overline{S}),\ \mathbf{0}}(\text{Depart})$. As shown in the proof of Lemma 10, this policy incurs a cost of at most

$2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}$ per transition. This gives us

$$\begin{aligned} V_\alpha(x) &\le V_{\alpha,\hat{\pi}}(x) \\ &\le \mathbb{E}\left[\sum_{t=1}^{X} \alpha^{t-1} \cdot 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*} + \alpha^X V_\alpha(\mathbf{0})\right] \\ &\le \mathbb{E}\left[\sum_{t=1}^{X} 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*} + V_\alpha(\mathbf{0})\right] \\ &\le \frac{2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}}{P^*_{\mathcal{L}(\overline{S}),\ \mathbf{0}}(\text{Depart})} + V_\alpha(\mathbf{0}), \end{aligned}$$

and so $h_\alpha(x)$ is bounded above as required.

SEN 3. We show that $0 \le h_\alpha(x)$. Showing that $0 \le h_\alpha(x)$ is equivalent to showing that $V_\alpha(\mathbf{0}) \le V_\alpha(x)$, which follows immediately from Lemma 9. ∎

Next, we show that it is never optimal to choose the Remain action indefinitely. Specifically, we show that there exists some finite number of orders that can be waiting at the depot where the optimal action is to depart, regardless of the destination of those orders.

Lemma 12 For any $\alpha \in (0, 1)$, let π^*_α be an α-discount optimal threshold policy as in Theorem 6. For all x ∈ X such that $\sum_{s\in S} x_s > 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}$, $\pi^*_\alpha(x) \neq$ *Remain.*

Proof Let $x \in X$ be given with $\sum_{s\in S} x_s > 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}$. Suppose for contradiction that $\pi^*_\alpha(x) =$ Remain. Then

$$V_\alpha(x) = \sum_{s\in S} x_s + \alpha(1 - \lambda\tau)V_\alpha(x) + \alpha \sum_{s\in S} f_S(s) V_\alpha(x + e^s).$$

From Lemma 9 we have that $V_\alpha(x + e^s) \ge V_\alpha(x)$, and so

$$\begin{aligned} V_\alpha(x) &\ge \sum_{s\in S} x_s + \alpha(1 - \lambda\tau)V_\alpha(x) + \alpha \sum_{s\in S} f_S(s) V_\alpha(x) \\ &\ge \sum_{s\in S} x_s + \alpha V_\alpha(x) \\ &\ge \frac{1}{1-\alpha} \sum_{s\in S} x_s. \end{aligned}$$

By assumption we have that $\sum_{s\in S} x_s > 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}$, and so

$$V_\alpha(x) > \frac{2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}}{1-\alpha}.$$

However, from Lemma 10 we know that $V_\alpha(x) \le 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}/(1-\alpha)$, a contradiction. ∎

Lemma 13 For any $\alpha \in (0, 1)$, let π^*_α be an α-discount optimal threshold policy as in Theorem 6. Then there exists $\alpha_1, \alpha_2, \ldots \in (0, 1)$ with $\lim_{n\to\infty} \alpha_n \to 1$ such that

$$\lim_{n\to\infty} \pi^*_{\alpha_n} \to \pi^*$$

exists where π^* is also a threshold policy.

Proof We characterize threshold policies by he states in which in the Remain action is taken. For any $\alpha \in (0, 1)$ define

$$\mathcal{R}_\alpha = \{x \in X | \pi^*_\alpha(x) = \text{Remain}\}.$$

From Lemma 12 we have for any $\alpha \in (0, 1)$ that $\pi^*_\alpha(x) \neq$ Remain for all $x \in X$ such that $\sum_{s\in S} x_s > 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*}$. Define

$$\mathcal{R} = \left\{ x \in X \,\middle|\, \sum_{s\in S} x_s \le 2\lambda\Pi(\mathcal{L}(\overline{S}))_{s^*} \right\}.$$

This is a finite set, and by construction $\mathcal{R}_\alpha \subseteq \mathcal{R}$. Let a sequence $\alpha_1, \alpha_2, \ldots \in (0, 1)$ with $\lim_{n\to\infty} \alpha_n \to 1$ be given. Then the sequence $\mathcal{R}_{\alpha_1}, \mathcal{R}_{\alpha_2}, \ldots$ is contained in the finite set $\mathcal{R}$ and so has some convergent subsequence $\mathcal{R}_{\beta_1}, \mathcal{R}_{\beta_2}, \ldots$ that converges to an element of $\mathcal{R}$, say $\mathcal{R}^*$. Thus

$$\lim_{n\to\infty} \pi^*_{\beta_n} \to \pi^*$$

exists and is a threshold policy, as required. ∎

We conclude by applying Theorem 7 to the threshold policy π^* of Lemma 13, which satisfies the conditions of the Theorem by Lemma 11, to conclude that π^* is an optimal threshold policy for the MDP Φ.

4 | CONCLUSION

We have analyzed the problem in adversarial and in average-case settings. In the adversarial case we presented a randomized algorithm that is $(3\beta\Delta/2\delta - 1)$-competitive against an oblivious adversary and is $(2\beta\Delta/\delta - 1)$-competitive against an adaptive offline adversary. We demonstrate that these competitive ratios are tight. Additionally, we showed that no online algorithm has a competitive ratio less than $1 + 0.271\Delta/\delta$. In the average-case setting we show that optimal policies exhibit a threshold structure.

The online algorithm we present in the adversarial setting has the advantage of being very easy to implement, as it requires only computing an approximate TSP tour and drawing a random number. The threshold policies we describe in the average-case setting can be more expensive to compute. For example, the work required to compute these policies using value iteration as in Sennott (2009) grows with the size of the state space $\mathbb{Z}_+^{|S|}$; even truncating the state space by considering only states with finitely many orders still has the size

of the state space grow exponentially with the number of locations in the metric space $\mathcal{S}$. We conjecture that algorithm 1 can perform nearly as well as the optimal threshold policies in some average-case settings. This is motivated by the structure of the adversarial inputs we used in the proof of Theorem 5, which are extremely unlikely to occur in the average-case setting. We offer numerical experiments in a simple geometry to support this conjecture in Appendix A1.

We make a number of assumptions that limit the applicability of our work to the real world. Below are a number of extensions to our problem that would make the results more practical.

4.1 | Capacitated couriers

We assume that the courier can travel with an unlimited number of orders from the depot. In reality, couriers will have only finite carrying capacity. For example, there could be limits on the weight, physical volume, or quantity of orders that can be handled by a courier on one excursion from the depot.

4.2 | Limited travel time

We assume that the courier can travel indefinitely without penalty. In reality, human operators can only navigate for so long without rest, and even automated systems need periodic refueling or maintenance.

4.3 | Order deadlines

We seek to minimize the total latency across all orders, but allow orders to be delivered at any point after they arrive. In reality, many situations either have hard constraints on delivery deadlines, or incur additional penalties if an order is delivered later than agreed upon. In these cases it might be natural to model the cost of delivering an order to be fixed if the order is delivered before the deadline expires, and incur some penalty otherwise.

4.4 | Multiple couriers

We assume that we are serving orders with only a single courier. Extending this work to allocate orders to a fleet of couriers would make this problem substantially more practical, as many real-world logistics problems are far too large for a single courier to handle.

ORCID

Patrick R. Steele http://orcid.org/0000-0002-0611-0206
Shane G. Henderson https://orcid.org/0000-0003-1004-4034

REFERENCES

Amazon flex. (2005). *Amazon Flex*. Retrieved from https://flex.amazon.com

Ascheuer, N., Krumke, S. O., & Rambau, J. (2000). *Online dial-a-ride problems: Minimizing the completion time*. In Annual Symposium on Theoretical Aspects of Computer Science (pp. 639–650). Berlin, Heidelberg: Springer.

Ausiello, G., Feuerstein, E., Leonardi, S., Stougie, L., & Talamo, M. (2001). Algorithms for the on-line travelling salesman. *Algorithmica*, *29*(4), 560–581.

Borodin, A., & El-Yaniv, R. (1998). Online computation and competitive analysis. New York, USA: Cambridge University Press.

Christofides, N. (1976a). *Worst-case analysis of a new heuristic for the travelling salesman problem*. Technical report, DTIC Document.

Christofides, N. (1976b). The vehicle routing problem. *Revue française d"automatique, d"informatique et de recherche opérationnelle. Recherche opérationnelle*, *10*(1), 55–70.

Cordeau, J.-F., & Laporte, G. (2007). The dial-a-ride problem: Models and algorithms. *Annals of Operations Research*, *153*(1), 29–46.

Feuerstein, E., & Stougie, L. (2001). On-line single-server dial-a-ride problems. *Theoretical Computer Science*, *268*(1), 91–105.

Irani, S., Lu, X., & Regan, A. (2004). On-line algorithms for the dynamic traveling repair problem. *Journal of Scheduling*, *7*(3), 243.

Jaillet, P., & Xin, L. (2014). Online traveling salesman problems with rejection options. *Networks*, *64*(2), 84–95.

Krumke, S. O., de Paepe, W. E., Poensgen, D., & Stougie, L. (2003). News from the online traveling repairman. *Theoretical Computer Science*, *295*(1), 279–294.

Sennott, L. I. (2009). Stochastic dynamic programming and the control of queueing systems (Vol. *504*). New York, USA: John Wiley & Sons.

Sitters, R. (2014). *Polynomial time approximation schemes for the traveling repairman and other minimum latency problems*. In Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (pp. 604–616). Philadelphia, USA: SIAM.

Uber rush. (2005). *How It Works*. Retrieved from https://rush.uber.com/how-it-works

Williamson, D. P., & Shmoys, D. B. (2011). The design of approximation algorithms. Cambridge, IL: Cambridge University Press.

How to cite this article: Steele P, Henderson S, Shmoys D. Aggregating courier deliveries. *Naval Research Logistics.* 2018;65:187–202. https://doi.org/10.1002/nav.21804

APPENDIX A: COMPARING THE TWO SETTINGS:

Computing an optimal threshold policy π^* for the CTMDP Ψ can be computationally expensive, whereas running algorithm 1 on a given input requires only evaluating the β-approximation $\text{TSP}_\beta(\mathcal{S})$. For this reason we are interested

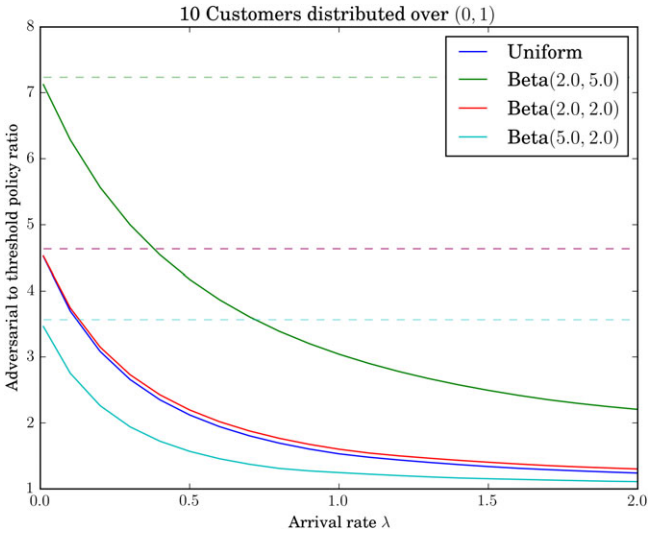

FIGURE A1 The ratio of the performance of algorithm 1 relative to the optimal threshold policy for different arrival rates. The solid lines indicate the ratio for different arrival distributions. The associated dashed lines represent the worst-case ratio for any arrival rate. The worst-case guarantees are easily computed by evaluating the expected cost per order of algorithm 1, which depends only on the mean order distance and not the arrival rate. The associated lower bound on the cost of optimal threshold policy is obtained by the structure of the cost functions

in bounding the performance of algorithm 1 relative to the performance of π^*, an optimal policy for Ψ, in the long-run average cost setting of section 3. We consider the simple geometry where S is a finite subset of $[0, \Delta/2]$ where the depot is at 0; note that by construction the optimal TSP tour will depart from the depot, move to the furthest customer at a distance $\Delta/2$, and then return, incurring a total distance of Δ. In this case we can consider $\beta = 1$ since the optimal tour is known.

In this simple geometry, the CTMDP of section 3 can be reduced to a much smaller state space. In particular, we need only track the total number of orders waiting for delivery, and the distance of the furthest customer to which there is an order. With these two pieces of information the tour we take when we depart and the costs incurred are known. To construct this equivalent CTMDP we move all delivery charges from the depart action to the remain action, noting that over any sequence of remain actions followed by a depart action we incur the same cost. We can then use any solution method to compute the long-run average cost incurred. We utilized the value iteration method from Sennott (2009).

Figure A1 shows the ratio between the performance of algorithm 1 relative to the optimal threshold policy when $S = \{0, 1/11, 2/11, \ldots, 10/11\}$. We considered uniform and (discretized) Beta distributions for the customer distribution f_S. As shown, for any arrival rates the performance guarantees are quite good. Note that in this setting $\beta = 1$, $\Delta = 20/11$, and $\delta = 1/11$, giving us a competitive ratio of 29 from theorem 1, which is far worse than any of these performance guarantees. This suggests that algorithm 1 can perform well in place of using a difficult-to-compute optimal threshold policy in certain settings.