Operations Research
Vol. 56, No. 3, May–June 2008, pp. 745–757
ISSN 0030-364X | EISSN 1526-5463 | 08 | 5603 | 0745

DOI 10.1287/opre.1070.0450

Generalized Online Routing: New Competitive Ratios, Resource Augmentation, and Asymptotic Analyses

Patrick Jaillet
Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, jaillet@mit.edu

Michael R. Wagner
Department of Management, California State University East Bay, Hayward, California 94542, michael.wagner@csueastbay.edu

We consider online routing optimization problems where the objective is to minimize the time needed to visit a set of locations under various constraints; the problems are online because the set of locations are revealed incrementally over time. We consider two main problems: (1) the online traveling salesman problem (TSP) with precedence and capacity constraints, and (2) the online TSP with m salesmen. For both problems we propose online algorithms, each with a competitive ratio of 2; for the m-salesmen problem, we show that our result is best-possible. We also consider polynomial-time online algorithms.

We then consider resource augmentation, where we give the online servers additional resources to offset the powerful offline adversary advantage. In this way, we address a main criticism of competitive analysis. We consider the cases where the online algorithm has access to faster servers, servers with larger capacities, additional servers, and/or advanced information. We derive improved competitive ratios. We also give lower bounds on the competitive ratios under resource augmentation, which in many cases are tight and lead to best-possible results.

Finally, we study online algorithms from an asymptotic point of view. We show that, under general stochastic structures for the problem data, unknown and unused by the online player, the online algorithms are almost surely asymptotically optimal. Furthermore, we provide computational results that show that the convergence can be very fast.

Subject classifications: online optimization; transportation; analysis of algorithms.
Area of review: Transportation.
History: Received May 2006; revisions received November 2006, February 2007; accepted April 2007. Published online in *Articles in Advance* January 4, 2008.

1. Introduction

The traveling salesman problem (TSP) is a very important problem in operations research; TSP solutions are valuable in their own right as well as in the solution of more complicated problems. In a common version of the TSP, we are given a metric space and a set of points in the space, representing cities. Given an origin city, the task is to find a tour of minimum total length, beginning and ending at the origin, that visits each city at least once. Assuming a constant speed, we can interpret this objective as minimizing the time required to complete a tour. We may also incorporate release dates, where a city must be visited on or after its release date. In this case, the problem is known as the "TSP *with release dates.*"

Additional constraints can be added to the above salesman problems. We will consider several in this paper. The salesman can be considered a vehicle/server that transports packages and/or people. We can introduce precedence constraints where some cities must be visited before others. Precedence constraints are appropriate, for example, if packages/people have to be picked up at one location and delivered to another location. It is also natural to introduce a capacity for the server; in other words, a server can visit only a subset of all cities in a given tour and must traverse multiple tours. Finally, we consider the case where we have multiple servers to manage. In this paper, we are concerned with online versions of the above mentioned routing optimization problems. In our framework, the problem data are revealed dynamically over time, independent of the server's location, at release dates.

It is well established that the assumption that problem instances are completely known a priori is unrealistic in many applications. There exist numerous approaches for solving optimization problems under uncertainty. Assuming a probabilistic distribution or process for the problem data leads to stochastic and dynamic programming formulations. However, this approach generally requires accurate probabilistic distributions. In practice, there might not be sufficient data to estimate these distributions accurately. This is the case particularly when one considers a new market or industry, or even when a known industry is in a period of upheaval. Therefore, more conservative approaches are needed. A popular approach to optimization under uncertainty is robust optimization (e.g., see Ben-Tal and Nemirovski 2000). The basic framework of robust optimization is to introduce sets for uncertain problem parameters rather than distributions. Attempts to make robust optimization less conservative have also been successful (e.g., see Bertsimas and Sim 2003). Our approach, online optimization, is a different approach to optimization under uncertainty, which is more appropriate for sequential decision-making problems where probabilistic distributions are not available. For example, taxi services, buses, and courier services require an online model in which locations to be visited are revealed over time, while the server is en route serving previously released requests. Because online optimization is also a conservative approach, we do make attempts to relax this aspect in the form of resource augmentation.

The focus of this paper is on studying algorithms for a variety of online routing problems. They are evaluated using the competitive ratio criteria, which is defined as the worst-case ratio of the online algorithm's cost to the cost of an optimal offline algorithm, where all data are known a priori. We also say that an algorithm is c-competitive if the competitive ratio of the algorithm is at most c. We call an online algorithm (or competitive ratio) *best-possible* if there does not exist another online algorithm with a strictly smaller competitive ratio. We provide online algorithms for new online routing problems, and we derive new competitive ratio bounds. A number of our competitive ratio results are best-possible.

The competitive ratio is a conservative worst-case measure, but it does provide a guarantee of a minimal level of performance. From a practitioner's point of view, a competitive ratio can provide a benchmark from which to compare other solution approaches. We do not claim that our approach is appropriate for solving all routing problems under uncertainty, but we do believe it is appropriate for problems where there is little information to characterize the uncertainty. For example, suppose that FedEx has expanded to serve (i.e., a specialized service such as same day delivery) a new market, perhaps in a new city or simply in a new neighborhood. Initially, no data are available for this new market, and online optimization is a valid approach. Once a sufficient amount of data is collected, a stochastic optimization approach could then be more appropriate. The research in this paper also serves as a starting point for routing optimization problems under partial uncertainty. For example, there might exist an underlying distribution for the problem data, but we know only the mean of the distribution. Similar approaches have been taken in the revenue management and supply chain literature; see Ball and Queyranne (2006), Lan et al. (2007), and Perakis and Roels (2006a, b).

We also study resource augmentation, where we give the online algorithm additional resources with respect to the offline algorithm. From a theoretical point of view, resource augmentation allows us to preclude the (usually) pathological worst-case examples that induce competitive ratios. With resource augmentation in place, we derive improved competitive ratios. We consider speed, capacity, server, and information augmentation. The information augmentation that we consider is in the form of advanced notice. In other words, the online algorithm receives a "heads up" that a request is coming in the near future. For example, a customer can call for a taxi at 3 P.M. and request a pickup at 3:30 P.M. The half-hour difference between the customer's calling a taxi and the pickup is the advanced notice. We quantify the improvement in competitive ratio as a function of requests' advanced notice. The other forms of resource augmentation that we consider allow us to quantify the value of adding additional resources. For example, our results allow us to answer the following questions: How much can we improve our worst-case performance if we add vehicles to our fleet or increase the capacity of our vehicle?

We also derive lower bounds for competitive ratios with or without resource augmentation. In many cases, our lower bounds are tight, implying best-possible competitive ratios. Finally, we investigate the asymptotic behavior of online algorithms under probabilistic assumptions. We show that many online algorithms are almost surely asymptotically optimal. We complement these theoretical results with computational studies, which show that the convergence to optimality can be very fast. From a practical point of view, we can say that if our problem instance is large enough, the additional cost of handling a dynamic optimization problem under uncertainty, compared to having all information a priori, is negligible. Furthermore, our computational studies suggest that "large enough" can actually be quite small: For 20 cities in the standard online TSP model, our computed online cost is at most 10% higher than optimal.

1.1. Literature Review

Research concerning online versions of the TSP have been introduced relatively recently. Most related to our paper is the stream of works that started with the paper by Ausiello et al. (2001). In this paper, the authors studied the online TSP, which is a special case of the problems we consider here. They analyzed the problem on the real line and on general metric spaces, developing online algorithms for both cases and achieving a best-possible online algorithm for general metric spaces, with a competitive ratio of 2. These authors also provide a polynomial-time online algorithm, for general metric spaces, which is 3-competitive. Subsequently, the paper by Ascheuer et al. (2000) implies the existence of a polynomial-time algorithm for general metric spaces, which is 2.65-competitive, as well as a $(2+\epsilon)$-competitive ($\epsilon > 0$) algorithm for Euclidean spaces.

There has also been work on generalizing the basic online TSP framework. The paper by Feuerstein and Stougie (2001) considers the online dial-a-ride problem, where each city is replaced by an origin-destination pair. The authors consider both the uncapacitated case, giving a best-possible 2-competitive algorithm, and the capacitated case, giving a 2.5-competitive algorithm. The paper by Ascheuer et al. (2000) also gives a 2-competitive online algorithm and a $(1+\sqrt{1+8\rho})/2$-competitive polynomial-time online algorithm for the uncapacitated online dial-a-ride problem (ρ being the approximation ratio of a simpler but related offline problem). Their algorithm is generalizable to the case where there are multiple servers with capacities; this generalization is also 2-competitive. Other groups of researchers have generalized the online TSP in other ways: Ausiello et al. (2005) have studied the online asymmetric TSP, Ausiello et al. (2004) have studied the online quota TSP, and Blom et al. (2001) have considered the online TSP under different adversarial models.

There has been limited work in multiple vehicle online routing problems. As mentioned previously, Ascheuer et al. (2000) give a 2-competitive online algorithm for the online dial-a-ride problem with multiple servers and capacity constraints. Bonifaci and Stougie (2007) study the online TSP with m salesmen. For the case where all cities are on the real line, these authors give an asymptotically (as $m \to \infty$) optimal online algorithm. These authors also focus on resource augmentation with respect to the number of vehicles: The online algorithm has m salesmen, and the offline algorithm has $m^* \leqslant m$ salesmen. These authors give an online algorithm that is $(1+\sqrt{1+1/2^{\lfloor m/m^* \rfloor -1}})$-competitive. Ausiello et al. (2006) also consider the behavior of online routing algorithms as a function of the number of servers.

In general, resource augmentation for online problems was introduced by Sleator and Tarjan (1985), who show that it is possible for an online paging algorithm to have a constant competitive ratio if it is given a constant fraction more cache locations than the offline algorithm. Server resource augmentation was considered by Young (1994) for the k-server problem and by Kalyanasundaram and Pruhs (2000a) for the online weighted-matching problem. Kalyanasundaram and Pruhs (2000b) consider speed and processor augmentation in online machine scheduling. In Jaillet and Wagner (2006), information augmentation is present in the form of *disclosure dates*; a similar approach was taken by Allulli et al. (2005) in the form of a *lookahead*. Other frameworks for addressing the limitations of the competitive ratio have also been introduced; see Ben-David and Borodin (1994), Koutsoupias and Papadimitriou (2000), and Raghavan (1992).

Studies considering the asymptotic analysis of online routing problems have been very limited. Hiller (2005) performs an asymptotic probabilistic competitive analysis of an online dial-a-ride problem on trees. A number of asymptotic optimality results have been given in the online machine scheduling literature; e.g., see Chou et al. (2006a, b) and Liu et al. (2005).

1.2. Our Contributions

We first consider single-server online routing problems with precedence and capacity constraints. We give an online algorithm that is 2-competitive; the power of this statement is that adding general precedence and capacity constraints to the online TSP does not increase the competitive ratio. Our result can be contrasted with a result in Ascheuer et al. (2000), which gives a 2-competitive online algorithm for the online dial-a-ride problem with multiple servers and capacity constraints. This result is more general than ours because it combines multiple servers with capacity constraints. However, in a different sense, this result is also simpler than ours because it considers only an origin-destination precedence constraint (our precedence constraints are more general) as well as unit demands (we allow for arbitrary demands). Considering polynomial-time algorithms, a modification to our algorithm is 2ρ-competitive, where ρ is the approximation ratio of a simpler offline problem. Ascheuer et al. (2000) give an improved result, a $(1+\sqrt{1+8\rho})/2$-competitive algorithm, but for the simpler online dial-a-ride problem with one server and no capacity constraints.

Next, we study multiple-server routing problems (without precedence and capacity constraints) and show similar results to those just mentioned. We design a new algorithm with a competitive ratio of 2, a result that is best-possible. This matches a 2-competitive algorithm that is implied in Ascheuer et al. (2000), which is for the more general online dial-a-ride problem. A modification of our algorithm again results in a polynomial-time online algorithm that is 2ρ-competitive. We are not aware of any other polynomial-time algorithms for online multiple-vehicle routing problems. We note that adding servers to the problem statement does not increase or decrease the competitive ratio with respect to the online TSP.

Next, we consider resource augmentation. For single-server problems with speed augmentation (with both precedence and capacity constraints), we show our polynomial-time online algorithm is $(1+(2\rho-1)/\gamma)$-competitive, where $\gamma \geqslant 1$ is the speed of the online server (the offline server moves at unit speed). If we do not restrict our algorithms to run in polynomial time, we can show the competitive ratio of our algorithm is $1+1/\gamma$, and this result is best-possible. We are not aware of any other similar results in the literature. For multiple-server problems, we consider speed and server augmentation (without precedence and capacity constraints); we show that our algorithm is $(1+\rho(1-(m-1)\phi)/\gamma+(\rho-1)/\gamma)$-competitive, where m is the number of online servers (the offline has a single server) and ϕ is a measure of the problem data. The reader may constrast our result with the vehicle augmentation result of Bonifaci and Stougie (2007). When the online algorithm has m servers and the offline algorithm has $m^* \leqslant m$ servers, the authors give an online algorithm that is $(1+\sqrt{1+1/2^{\lfloor m/m^* \rfloor -1}})$-competitive. This result is more general than ours in the sense that they allow the offline to have any number of servers, not just a single server as in our case. Furthermore, they do not require a measure of the problem data (ϕ) as we do. However, our result is more general than theirs because we can simultaneously show the impact of speed augmentation for polynomial-time algorithms. Comparing the two results equitably (setting $\gamma = \rho = m^* = 1$), our competitive ratio bound is $(2-(m-1)\phi)$ and theirs is $(1+\sqrt{1+1/2^{m-1}})$. If both online and offline algorithms have m servers at their disposal, our algorithm is again $(1+(2\rho-1)/\gamma)$-competitive. If we do not restrict our algorithms to run in polynomial time, we can show the competitive ratio of our algorithm is $1+1/\gamma$, and this result is best-possible. Finally, we consider the notion of advanced information, where a city is revealed to the online algorithm before its release date. We give a result for the multiple-server case that generalizes the single-server result in Jaillet and Wagner (2006): We give an algorithm that is $(2-\alpha/(1+\alpha))$-competitive, where α is a measure of the advanced information. This improves the best-possible competitive ratio of 2 when there is no advanced information.

In the final part of our paper, we study the asymptotic behavior of single-server online routing under general stochastic structures for the problem data, *unknown and unused by the online algorithm*. We first consider capacity and speed augmentation (with no precedence constraints) when city locations are stochastic (with arbitrary release dates and unit demands); we show that our polynomial-time online algorithm is asymptotically $(1+(\rho q)/(\gamma Q)+(\rho-1)/\gamma)$-competitive, almost surely, where $Q(q)$ is the capacity of the online (offline) server ($Q \geqslant q$). We then consider the case where both online and offline algorithms have the same resources but where city locations, release dates, and—at times—demands are stochastic. We give online algorithms that are almost surely asymptotically optimal for two frameworks: (1) precedence constraints but no capacity constraints, and (2) capacity constraints but no precedence constraints. We are aware of only one other related result: Hiller (2005) considers the online dial-a-ride problem on trees under high loads and gives an online algorithm that is probabilistically $(1+o(1))$-competitive. Similar asymptotic optimality results can be found in the online machine scheduling literature; see Chou et al. (2006a, b) and Liu et al. (2005). Finally, we provide a computational study that shows that the convergence to optimality can be very fast.

Outline. The remainder of this paper is organized as follows. In §2, we give preliminaries and study general online routing problems using the traditional competitive ratio measure. In §3, we study the effect of resource augmentation for a variety of problems. In §4, we show the almost sure asymptotic optimality of a number of online routing algorithms, and we provide computational results on the rate of convergence. Finally, in §5 we provide concluding remarks.

2. Generalized Online Routing

2.1. Single-Server Routing Problems

We first consider routing problems where a single server must service a sequence of requests. The data for our problems are a set of points $(\mathbf{l}_i, r_i, \mathbf{d}_i)$, $i=1,\ldots,n$, where n is the number of requests and $k(i)$ is the number of cities in request i: $\mathbf{l}_i = (l_i^1, l_i^2, \ldots, l_i^{k(i)})$ and $\mathbf{d}_i = (d_i^1, d_i^2, \ldots, d_i^{k(i)})$. The quantity $l_i^j \in \mathcal{M}$, $\mathcal{M}$ an arbitrary metric space, is the location of the jth city in the ith request. The quantity $r_i \in \mathbb{R}_+$ is the ith request's release date; i.e., r_i is the first time after which cities in request i will accept service. We assume, without loss of generality, that $r_1 \leqslant r_2 \leqslant \cdots \leqslant r_n$. The quantity $d_i^j \in \mathbb{R}_+$ is the demand of city l_i^j. The server has a capacity Q and the sum of city demands visited on any given tour can be at most Q; we assume that $d_i^j \leqslant Q$ for all i, j. It is possible to generalize our capacity model to allow positive and negative demands as well as different types of products being transferred. However, we study the current problem to limit the complexity of the analysis. Precedence constraints exist within a request; i.e., for a fixed i, arbitrary precedence constraints of the form $l_i^j \preceq l_i^k$ (l_i^j must be visited before l_i^k) for any $j \neq k$. The service requirement at a city is zero. Unless stated otherwise, the server travels at unit speed or is idle. The problem begins at time 0, and the server is initially at a designated origin o of the metric space. The objective is to minimize the time required to visit all cities and have the server return to the origin. We also let $\mathcal{N} = \{1, \ldots, n\}$.

From the online perspective, the total number of requests, represented by the parameter n, is not known, and request i only becomes known at time r_i. $Z^A(n, Q)$ denotes the cost of online algorithm A on an instance of n cities with server capacity Q, and $Z^*(n, Q)$ is the corresponding optimal offline cost where all data are known a priori. $Z^{r=0}(n, Q)$ is the optimal cost when all release dates are equal to zero; clearly, $Z^{r=0}(n, Q) \leqslant Z^*(n, Q)$. The problem instance underlying $Z^{r=0}(n, Q)$, $Z^A(n, Q)$, and $Z^*(n, Q)$ will be clear from context. Finally, define L_{TSP} as the optimal TSP tour length through all cities in an instance; i.e., $L_{\mathrm{TSP}} = Z^{r=0}(n, \infty)$; the value of n will be clear from context.

We measure the performance of online algorithms using the competitive ratio and the asymptotic competitive ratio. The competitive ratio is defined as the worst-case ratio, over all problem instances, of online to offline costs: $\max_{\text{instances}} Z^A(n, Q)/Z^*(n, Q)$. An online algorithm is also said to be c-competitive if its competitive ratio is at most c. An online algorithm is asymptotically c-competitive if there exists n_0 such that for all $n \geqslant n_0$, $Z^A(n, Q)/Z^*(n, Q) \leqslant c$. An online algorithm is said to be best-possible if there does not exist another online algorithm with a strictly smaller competitive ratio.

We first present online algorithm Plan-At-Home (PAH), which was given by Ausiello et al. (2001) and solves the online TSP optimally, with respect to the competitive ratio measure. We then generalize PAH and we denote our algorithm Plan-At-Home-Generalized (PAH-G). Note that the competitive ratio of the original PAH is 2.

ALGORITHM 1 (PAH)

(1) Whenever the server is at the origin, it calculates and implements an exact solution to L_{TSP} over all requests

whose release dates have passed but have not yet been served completely.

(2) If at time r_i, for some i, a new request is presented, the server takes one of two actions depending on the server's current position p and the current request location l_i:

(a) If $d(l_i, o) > d(p, o)$, the server goes back to the origin where it appears in a Case (1) situation.

(b) If $d(l_i, o) \leqslant d(p, o)$, the server ignores request i until it completes the route it is currently traversing, where again Case (1) is encountered.

Algorithm 2 (PAH-G)

(1) Whenever the server is at the origin, it calculates and implements a ρ-approximate solution to $Z^{r=0}(n, Q)$ over all requests whose release dates have passed but have not yet been served completely.

(2) If at time r_i, for some i, a new request is presented, the server takes one of two actions depending on the server's current position p and the farthest location in the current request l_i^*:

$$l_i^* = \underset{\{l_i^j \mid 1 \leqslant j \leqslant k(i)\}}{\arg\max}\ d(o, l_i^j):$$

(a) If $d(l_i^*, o) > d(p, o)$, the server goes back to the origin where it appears in a Case (1) situation.

(b) If $d(l_i^*, o) \leqslant d(p, o)$, the server ignores request i until it completes the route it is currently traversing, where again Case (1) is encountered.

We give a corollary of Theorem 1, which is proven in §3.

Corollary 1. *Algorithm PAH-G is* 2ρ*-competitive.*

As an example, if we consider the online capacitated TSP without precedence constraints, we can apply the iterated tour partition (ITP) heuristics given by Altinkemer and Gavish (1990) and Haimovich and Rinnooy Kan (1985). If $d_i^j = 1$ for all i, j, there exists an ITP heuristic with approximation ratio $\rho \leqslant (5/2 + 3/2Q)$. If demands are arbitrary, there exists an ITP heuristic with approximation ratio $\rho \leqslant (7/2 - 3/Q)$.

This result shows interesting properties. First, it is possible to relate the competitive ratio of PAH-G to the approximation ratio of a simpler but related optimization problem $Z^{r=0}(n, Q)$. Also, if we have access to exact algorithms for $Z^{r=0}(n, Q)$, adding capacity and precedence constraints results in no increase in the competitive ratio, with respect to the online TSP.

2.2. Multiple-Server Routing Problems

We now consider routing problems with m identical servers. We do not consider capacity or precedence constraints. The data for our multiple-server problems are closely related to that of the single-server problems: The data are a set of points (l_i, r_i), $i = 1, \ldots, n$, where $l_i \in \mathcal{M}$ ($r_i \in \mathbb{R}^+$) is the location (release date) of the ith request. We again assume, without loss of generality, that $r_1 \leqslant r_2 \leqslant \cdots \leqslant r_n$. The service requirement at a city is again zero. Unless stated otherwise, the servers travel at unit speed or are idle. The problem begins at time 0, and all servers are initially at a designated origin o of the metric space. The objective is to minimize the time required to visit all cities and have all servers return to the origin.

$Z^A(n, m)$ denotes the cost of online algorithm A on an instance of n cities with m identical servers, and $Z^*(n, m)$ is the corresponding optimal offline cost where all data are known a priori. We assume that $n \geqslant m$. $Z^{r=0}(n, m)$ is the optimal cost when all release dates are equal to zero; clearly, $Z^{r=0}(n, m) \leqslant Z^*(n, m)$. Note that $Z^{r=0}(n, m)$ is equivalent to the problem of finding a set of m tours, that collectively visit all locations, such that the maximum tour length is minimized; see Frederickson et al. (1978). The problem instance underlying $Z^{r=0}(n, m)$, $Z^A(n, m)$, and $Z^*(n, m)$ will be clear from context. Finally, note that $L_{\text{TSP}} = Z^{r=0}(n, 1)$; the value of n will be clear from context. The competitive ratio and (asymptotic) competitiveness are defined similarly to the single-server case.

We again give an online algorithm that generalizes PAH, which was given by Ausiello et al. (2001); we denote our algorithm Plan-At-Home-m-Servers (PAH-m). Let p_i denote the location of server i.

Algorithm 3 (PAH-m)

(1) Whenever all servers are at the origin, they calculate and implement a ρ-approximate solution to $Z^{r=0}(n, m)$ over all requests whose release dates have passed but have not yet been served.

(2) If at time r_i, for some i, a new request is presented, the servers take one of two actions depending on the request's location l_i and the farthest server's current position p^* (ties broken arbitrarily):

$$p^* = \underset{\{p_i \mid 1 \leqslant i \leqslant m\}}{\arg\max}\ d(o, p_i):$$

(2a) If $d(l_i, o) > d(p^*, o)$, all servers go back to the origin where they appear in a Case (1) situation.

(2b) If $d(l_i, o) \leqslant d(p^*, o)$, all servers except p^* return to the origin; server p^* ignores request i until it completes the route it is currently traversing, where again Case (1) is encountered.

We first give a corollary of Theorem 4, and then we give a corollary of Theorem 5. These theorems are proven in §3.

Corollary 2. *Algorithm PAH-m is* 2ρ*-competitive.*

As an example, we can apply the approximation algorithm for $Z^{r=0}(n, m)$ given by Frederickson et al. (1978) that has an approximation ratio $\rho \leqslant 5/2 - 1/m$.

Corollary 3. *If we use an exact algorithm in Step* (1) *for calculating an optimal offline* $Z^{r=0}(n, m)$, *the competitive ratio of PAH-m is* 2 *and this result is best-possible.*

Again, it is possible to relate the competitive ratio to the approximation ratio of a simpler but related optimization problem $Z^{r=0}(n,m)$. Also, if we have access to exact offline algorithms for $Z^{r=0}(n,m)$, adding extra vehicles to the problem statement results in no change (increase or decrease) in the competitive ratio, with respect to the online TSP.

3. Resource Augmentation

3.1. Definition

Resource augmentation gives the online algorithm additional power by increasing its resources. The motivation is to preclude pathological examples that could drive the worst-case competitive ratio; with resource augmentation, we derive improved, more realistic, and meaningful competitive ratios.

We study four types of resource augmentation: speed, capacity, server, and advanced notice. Speed augmentation gives the online servers, a faster speed than the corresponding offline servers. Capacity augmentation gives the online server a larger capacity. Vehicle augmentation gives the online algorithm more servers than the offline algorithm. Advanced notice allows each request to be revealed to the online algorithm ahead of its release date. In §3.2, we first consider our general single-server routing framework with speed augmentation (in §4.1, with additional stochastic assumptions, we consider single-server problems with both speed and capacity augmentation). We then consider in §3.3 multiple-server routing problems with both speed and vehicle augmentation. In §3.4, we extend the single-server advanced notice results in Jaillet and Wagner (2006) to the multiple-server case. Finally, in §3.5 we present lower bounds for competitive ratios under resource augmentation. These results are used to show that a number of our competitive ratio results are best-possible.

3.2. Single-Server Resource Augmentation

The online algorithm has a single server with a speed $\gamma \geqslant 1$. The offline algorithm has a single server of unit speed. Online and offline servers have identical capacities Q.

THEOREM 1. *Algorithm PAH-G is* $(1+(2\rho-1)/\gamma)$*-competitive.*

PROOF. See the online appendix. An electronic companion to this paper is available as part of the online version that can be found at http://or.pubs.informs.org/ecompanion.html. □

We also have a tight result when we have access to exact offline algorithms.

THEOREM 2. *If we use an exact algorithm in Step* (1) *for calculating an optimal offline* $Z^{r=0}(n,Q)$, *the competitive ratio of PAH-G is* $1+1/\gamma$, *and this result is best-possible.*

PROOF. We give matching upper and lower bounds on the competitive ratio. The upper bound of $1+1/\gamma$ is clear by setting $\rho=1$ in Theorem 1. The lower bound is clear by setting $a=0$ in Theorem 8. □

3.3. Multiple-Server Resource Augmentation

In this subsection, we give the online algorithm m identical servers, each with a speed $\gamma \geqslant 1$. The offline has a single server of unit speed.

Next, we make an observation: If all the locations are closely clustered, there is little benefit to using multiple vehicles. Therefore, we make an assumption that allows us to circumvent this fact; we then give a useful lemma that applies this assumption.

ASSUMPTION 1. *There exists* $\beta>0$ *such that for all* $i,j \in \{0,\dots,n\}$, $i\neq j$,

$$d(l_i,l_j)\geqslant \beta,$$

where l_0 *denotes the origin.*

LEMMA 1. $Z^{r=0}(n,m)\leqslant Z^{r=0}(n,1)-(m-1)\beta$.

PROOF. We consider a feasible solution to $Z^{r=0}(n,m)$: For servers $1,\dots,m-1$, we assign them each the $(m-1)$ locations closest to the origin. For vehicle m, we assign it the remaining $n-m+1$ locations; clearly $Z^{r=0}(n,m)$ equals the distance traveled by server m. Because distances between locations are at least β, server m will require at most a distance of $L_{\mathrm{TSP}}-(m-1)\beta$ to serve the remaining locations. □

Next, we need to define a measure of the value of the lower bound β. Let

$$\phi=\frac{\beta}{L_{\mathrm{TSP}}};$$

note that because $L_{\mathrm{TSP}}\geqslant (n+1)\beta$, we have that $\phi\leqslant 1/(n+1)$; there exist instances such that this last inequality is tight, the most obvious being the metric space where $d(l_i,l_j)=\beta$ for all $i\neq j$.

THEOREM 3. *Under Assumption* 1, *the competitive ratio of Algorithm PAH-m is at most*

$$1+\frac{\rho}{\gamma}(1-(m-1)\phi)+\frac{\rho-1}{\gamma}.$$

PROOF. See the online appendix. □

THEOREM 4. *If both online and offline algorithms have m servers, the competitive ratio of Algorithm PAH-m is at most*

$$1+\frac{2\rho-1}{\gamma}.$$

PROOF. Repeat the proof of Theorem 3 but replace all instances of $Z^*(n,1)$ with $Z^*(n,m)$ (all bounds remain valid) and do not apply Lemma 1. □

THEOREM 5. *If both online and offline algorithms have m servers and the online algorithm has access to an exact algorithm in Step* (1) *for calculating an optimal offline* $Z^{r=0}(n,m)$, *the competitive ratio of PAH-m is* $1+1/\gamma$, *and this result is best-possible.*

Proof. We give matching upper and lower bounds on the competitive ratio. The upper bound of $1+1/\gamma$ is clear by setting $\rho = 1$ in Theorem 4. The lower bound is clear by setting $a = 0$ in Theorem 8. □

A stronger (although dependent on n) parallel development is also possible. We first give an improvement to Lemma 1.

Lemma 2. $Z^{r=0}(n,m) \leqslant Z^{r=0}(n,1) - (n(m-1)/m)\beta$.

Proof. For simplicity, assume that $n/m \in \mathbb{N}$. Let $l = n \cdot (m-1)/m$. Note that l satisfies the following equation: $l/(m-1) \leqslant n-l$. Next, we assign $l/(m-1)$ locations to each of the first $(m-1)$ servers and then assign $(n-l)$ to server m. We do this in the following way: Pick the $l/(m-1)$ locations that form the shortest tour and assign this tour to server 1. Out of the remaining locations, find the next $l/(m-1)$ locations that form the shortest tour and assign to server 2. Repeat until vehicle $(m-1)$. Therefore, vehicle m will have the longest tour of all vehicles. Consequently, $Z^{r=0}(n,m) \leqslant Z^{r=0}(n,1) - l\beta = Z^{r=0}(n,1) - (n(m-1)/m)\beta$. □

Using this refinement, we have the following result.

Theorem 6. *Under Assumption* 1, *the competitive ratio of Algorithm PAH-m is at most*

$$1 + \frac{\rho}{\gamma}\left(1 - \frac{n(m-1)}{m}\phi\right) + \frac{\rho-1}{\gamma}.$$

3.4. Value of Advanced Information

In this section, we investigate the value of advanced information, as introduced by Jaillet and Wagner (2006), for the multiple-server case. In Jaillet and Wagner (2006), disclosure dates were introduced: q_i is the disclosure date of request i. We let q_i be the time when request i's data are revealed to the online algorithm; we require that $q_i \leqslant r_i$. We consider a special case where there exists a constant $a > 0$ such that $q_i = (r_i - a)^+$, where $(x)^+ = \max\{0, x\}$. We define an appropriate algorithm to take advantage of the disclosure dates, which we denote Plan-At-Home-m-Servers-disclosure-dates (PAH-m-dd).

Algorithm 4 (PAH-m-dd)

(1) Whenever all servers are at the origin, they calculate and implement an exact solution to $Z^*(n,m)$ over all requests whose disclosure dates have passed but have not yet been served completely.

(2) If at time q_i, for some i, a new request is presented, the servers take one of two actions depending on the request's location l_i and the farthest server's current position p^* (ties broken arbitrarily):

$$p^* = \underset{\{p_i \mid 1 \leqslant i \leqslant m\}}{\arg\max}\; d(o, p_i).$$

(a) If $d(l_i, o) > d(p^*, o)$, all servers go back to the origin where they appear in a Case (1) situation.

(b) If $d(l_i, o) \leqslant d(p^*, o)$, all servers except p^* return to the origin; server p^* ignores request i until it completes the route it is currently traversing, where again Case (1) is encountered.

Theorem 7. *Algorithm PAH-m-dd is* $(2-\alpha/(1+\alpha))$*-competitive, where* $\alpha = a/Z^{r=0}(n,m)$.

Proof. See the online appendix. □

3.5. Lower Bounds

In this subsection, we give two general lower bounds. The first considers the case where both online and offline algorithms have access to m servers. The second considers the case where the online algorithm has m servers and the offline algorithm has access to a single server.

Theorem 8. *When both online and offline algorithms have access to m servers, any* ρ*-competitive algorithm serving requests on a metric space* $\mathcal{M}$, *with* $q_i = (r_i - a)^+$, $i \in \mathcal{N}$ *and speed augmentation, has*

$$\rho \geqslant \frac{1+1/\gamma}{1+\alpha},$$

where $\alpha = a/Z^{r=0}(n,m)$.

Proof. See the online appendix. □

Theorem 9. *When the online algorithm has access to m servers and the offline algorithm a single server* (*i.e., vehicle augmentation*), *any* ρ*-competitive algorithm serving requests on a metric space* $\mathcal{M}$, *with* $q_i = (r_i - a)^+$, $i \in \mathcal{N}$ *and speed augmentation, has*

$$\rho \geqslant \frac{1+1/\gamma}{m+\alpha},$$

where $\alpha = a/Z^{r=0}(n,m)$.

Proof. See the online appendix. □

4. Asymptotic Analyses

In this section, we analyze online algorithms under stochastic structures for the problem data, unknown and unused by the online player. We use uppercase letters to denote random variables.

We consider only single-server problems, and we also require that each request be the same size: $\exists k$ such that $k(i) = k$ for all i. In §4.1, we first consider the special case $k = 1$, and we study capacity augmentation under stochastic assumptions on the request locations (release dates are arbitrary and demands unit); we are able to obtain a competitive ratio result beyond what we could achieve under the framework of §3. We then study in §4.2 the case where capacities are infinite but precedence constraints exist. In this section, we make stochastic assumptions on both the request locations and the release dates. Subsequently, in §4.3 we return to the special case $k = 1$ and study capacitated routing problems. We make stochastic assumptions on the request locations, release dates, and, at times, the request demands. In both §§4.2 and 4.3, we prove a number of almost sure asymptotic optimality results. Finally, in §4.4 we present computational results.

We utilize a generic technique to prove almost sure asymptotic optimality: We find random variables $F(n)$ and $G(n)$ that satisfy $Z^*(n,Q) \geqslant F(n)$ and $Z^A(n,Q) \leqslant F(n) + G(n)$ for all n for some online algorithm A. Then, we show that $\lim_{n\to\infty} G(n)/F(n) = 0$, a.s., which implies that $\lim_{n\to\infty} Z^A(n,Q)/Z^*(n,Q) = 1$, a.s.

We now list all the different assumptions that might be called upon throughout §4.

Spatial Stochastic Assumption

Assumption 2. *For each $j \in \{1,\ldots,k\}$, $L_1^j, L_2^j, \ldots, L_n^j$ are independently identically distributed from a distribution of compact support in $d \geqslant 2$ dimensional Euclidean space. Additionally, L_k^i and L_l^j are independent for all i, j, k, l (except, of course, when $i = j$ and $k = l$).*

Remark 1. Note that the distribution for $L_1^j, L_2^j, \ldots, L_n^j$ need not be the same as the distribution for $L_1^i, L_2^i, \ldots, L_n^i$ for $i \neq j$. The support for the individual distributions do not even need to overlap.

Temporal Stochastic Assumptions. We introduce two natural probabilistic structures for the release dates. We first consider a structure that is motivated by the uniformity of the requests and a second structure that is motivated by the common use of the Poisson process in modeling arrivals over time.

Assumption 3 (Order Statistics). *The release date of each request is a realization of a generic nonnegative random variable $Y \geqslant 0$; i.e., the unordered release dates are independently identically distributed from a given distribution. Because our model requires an order ($R_k \leqslant R_l$ for $k < l$), the kth release date is the kth order statistic: $R_k = Y_{(k)}$, where $Y_k \geqslant 0$, $k = 1, \ldots, n$ are i.i.d. random variables and $Y_{(1)} \leqslant Y_{(2)} \leqslant \cdots \leqslant Y_{(n)}$. Let μ_Y and σ_Y^2 denote the mean and variance, respectively, of the random variable Y.*

Assumption 4 (Renewal Process). *Define nonnegative i.i.d. random variables $X_i \geqslant 0$ to be the time between the $(i-1)$th and ith release date. We then define the release dates as follows: $R_k = \sum_{i=1}^k X_i$; note that $R_{k+1} = R_k + X_{k+1}$ for all k. Let μ_X and σ_X^2 denote the mean and variance, respectively, of the random variable X.*

Demand Stochastic Assumption. This assumption is used together with the normalization $Q = 1$.

Assumption 5. *The request demands D_i^j are i.i.d. from a distribution on $[0,1]$.*

4.1. Capacity Resource Augmentation for the Case $k = 1$ and $d_i = 1\ \forall i$

We let the online algorithm have a single server with a capacity of Q and a speed $\gamma \geqslant 1$. The offline algorithm has a single server of unit speed with capacity $q \leqslant Q$. Q and q are constants. Assumption 2 holds in this section. Request release dates are arbitrary and demands unit. We begin with a lemma.

Lemma 3. *Under Assumption 2, if $k = 1$, $0 < \mathbb{E}[d(0,L)] < \infty$ and $d_i = 1\ \forall i$, then $\lim_{n\to\infty} Z^{r=0}(n,Q)/Z^{r=0}(n,q) = q/Q$, a.s.*

Proof. Because Q and q are constants, we can apply the results of Haimovich and Rinnooy Kan (1985), which state that

$$\frac{Z^{r=0}(n,Q)}{n} \to \frac{2\mathbb{E}[d(0,l)]}{Q} \quad \text{and}$$

$$\frac{Z^{r=0}(n,q)}{n} \to \frac{2\mathbb{E}[d(0,l)]}{q}, \quad \text{a.s.}$$

Taking the limit of the ratio gives the result. □

Theorem 10. *Under Assumption 2, if $k = 1$, $0 < \mathbb{E}[d(0,L)] < \infty$ and $d_i = 1\ \forall i$, then the asymptotic competitive ratio of Algorithm PAH-G is at most $1 + \rho q/(\gamma Q) + (\rho - 1)/\gamma$, a.s.*

Proof. See the online appendix. □

4.2. Precedence Constraints Without Capacity Constraints

We consider here the general case $k \geqslant 1$. Note that when $k = 1$, we study the online TSP; and when $k = 2$, we study an online version of the makespan-objective dial-a-ride problem. We assume that $Q = \infty$. The precedence constraints are as follows: $\forall i$, $L_i^1 \preceq L_i^2 \preceq \cdots \preceq L_i^k$. In other words, L_i^1 must be visited before L_i^2, which in turn must be visited before L_i^3 and so on. Let L_{TSP}^j denote the shortest tour through the points $\{L_1^j, \ldots, L_n^j\}$ for each $j \in \{1,\ldots,k\}$. We begin by defining a greedy strategy which we denote as greedy-makespan (GM).

Algorithm 5 (GM)

At any release date, calculate a path $\mathcal{P}$ of shortest length that satisfies the following constraints:

(1) $\mathcal{P}$ starts at the current server location and ends at the origin o.

(2) All unserved requests are visited, and the precedence constraints are respected.

The server then traverses the path $\mathcal{P}$ at unit speed, until the next release date (if any).

Lemma 4.

$$Z^*(n,\infty) \geqslant \max\{R_n, \max_{1\leqslant j\leqslant k} L_{\text{TSP}}^j\} \quad \text{and}$$

$$Z^{\text{GM}}(n,\infty) \leqslant \min\left\{R_n + \frac{3}{2}\sum_{j=1}^k L_{\text{TSP}}^j,\ 2R_n + \sum_{j=1}^k L_{\text{TSP}}^j\right\}.$$

Proof. We first discuss the lower bounds on $Z^*(n,\infty)$. Clearly, $Z^*(n,\infty) \geqslant R_n$, the release date of the last request. Next, because for a given $j \in \{1,\ldots,k\}$ the locations $L_1^j,\ldots,L_n^j$ must be visited, the server must travel at least L_{TSP}^j, the shortest tour through these points. As the server travels at unit speed, $Z^*(n,\infty) \geqslant \max_{1\leqslant j\leqslant k} L_{\mathrm{TSP}}^j$.

We now consider strategy GM. At time R_n, the greedy server will optimize a path, from its current location through all remaining unserved points and finally returning to the origin. This greedy path will not take longer than the following alternate strategy: At time R_n, the server returns to the origin and then completes m tours in order. The first tour visits the points $L_1^1,\ldots,L_n^1$ and takes L_{TSP}^1 units of time, because the server travels at unit speed. Then, the server traverses L_{TSP}^2, followed by L_{TSP}^3 and so on, up to L_{TSP}^k. Clearly, this is a feasible strategy. Thus, strategy GM's cost may be bounded:

$$Z^{\mathrm{GM}}(n,\infty) \leqslant R_n + x + \sum_{j=1}^{k} L_{\mathrm{TSP}}^j, \tag{1}$$

where x is the time required for the inital return to origin at time R_n. Because the server travels at unit speed, clearly, $x \leqslant R_n$. Finally, the GM strategy will never allow the server to proceed past the maximum point location; this gives

$$x \leqslant \max_{1\leqslant j\leqslant k}\left\{\max_{1\leqslant i\leqslant n}\{d(o,L_i^j)\}\right\} \leqslant \max_{1\leqslant j\leqslant k}\left\{\tfrac{1}{2}L_{\mathrm{TSP}}^j\right\} \leqslant \frac{1}{2}\sum_{j=1}^{k} L_{\mathrm{TSP}}^j,$$

which gives the result. □

Note that GM is not a polynomial-time strategy, because for even $k=1$ we must calculate a Hamiltonian path. We now give a polynomial-time algorithm greedy-makespan-polynomial (GMP) that gives a result almost as strong as Lemma 4.

Algorithm 6 (GMP)

At any release date R_i:

(1) Return to the origin.

(2) For each $j \in \{1,\ldots,k\}$, calculate a tour using Christofides' heuristic to visit any unserved points in $\{L_1^j,\ldots,L_i^j\}$.

(3) Traverse the tours in the order $1,2,\ldots,k$ at unit speed, until the next release date (if any).

We have a corollary for algorithm GMP, which is easily seen by replacing Equation (1) in the proof of Lemma 4 with $Z^{\mathrm{GMP}}(n,\infty) \leqslant R_n + x + \frac{3}{2}\sum_{j=1}^{k} L_{\mathrm{TSP}}^j$.

Corollary 4.

$$Z^{\mathrm{GMP}}(n,\infty) \leqslant \min\left\{R_n + 2\sum_{j=1}^{k} L_{\mathrm{TSP}}^j,\ 2R_n + \frac{3}{2}\sum_{j=1}^{k} L_{\mathrm{TSP}}^j\right\}.$$

Finally, we consider algorithm PAH for the online TSP (i.e., $k=1$), which was defined by Ausiello et al. (2001) and generalized to be polynomial-time by Jaillet and Wagner (2006).

Algorithm 7 (PAH)

(1) Whenever the salesman is at the origin, it starts to follow a tour that serves all known cities that have not yet been served. This tour is calculated using a ρ-approximation algorithm that solves an offline classic TSP.

(2) If at time R_i, for some i, a new city is presented at point L, the salesman takes one of two actions depending on the salesman's current position p:

(a) If $d(L,o) > d(p,o)$, the salesman goes back to the origin where it appears in a Case (1) situation.

(b) If $d(L,o) \leqslant d(p,o)$, the salesman ignores the city until it arrives at the origin, where again it re-enters Case (1).

Using an argument similar to that in in the proof of Lemma 4, we attain the following result.

Lemma 5. $Z^{\mathrm{PAH}}(n,\infty) \leqslant R_n + 2\rho L_{\mathrm{TSP}}$.

Next, we study each of the release date structures. We consider only $k=1$ under the order statistic structure because our techniques were not successful for $k>1$. Fortunately, under the renewal process structure we were able to prove an asymptotic optimality result for any k. Furthermore, we show asymptotic optimality results for both polynomial-time algorithms GMP and PAH. The following theorem is essential to proving many of our asymptotic optimality results.

Theorem 11 (Beardwood et al. 1959). *Under Assumption* 2, *there exists a* $c_d^j > 0$ *such that*

$$\lim_{n\to\infty} \frac{L_{\mathrm{TSP}}^j}{n^{(d-1)/d}} = c_d^j \quad a.s,$$

where d *is the dimension of the underlying Euclidean space.*

4.2.1. Order Statistic Release Dates for $k=1$. For the order statistic release date structure, we consider only $k=1$. We begin by stating the main result of this subsection.

Theorem 12. *Under Assumptions* 2 *and* 3, *if* $k=1$ *and* $\sigma_Y^2 < \infty$, *then* $\lim_{n\to\infty}(Z^{\mathrm{GM}}(n,\infty)/Z^*(n,\infty)) = 1$ *a.s.*

To prove Theorem 12, we begin with a useful lemma concerning R_n.

Lemma 6. *If* $\mathbb{E}\{Y^r\} < \infty$, $r \in \mathbb{N}$, *then* $\lim_{n\to\infty} Y_{(n)}/n^{\delta} = 0$ *a.s., for any* $\delta \geqslant 1/r$.

Proof. Consequence of Theorem 4.4.1 in Galambos (1987). □

Proof of Theorem 12. We find random variables $F(n)$ and $G(n)$ such that $Z^*(n,\infty) \geqslant F(n)$ and $Z^{\mathrm{GM}}(n,\infty) \leqslant F(n)+G(n)$. We then prove that $\lim_{n\to\infty} G(n)/F(n) = 0$. Now, with $k=1$, Lemma 4 lets us define $F(n) = L_{\mathrm{TSP}}$ and $G(n) = 2R_n$. By Theorem 11, we have that there exists a $c_d > 0$ such that $\lim_{n\to\infty} F(n)/n^{(d-1)/d} = c_d$ a.s., and consequently, $\lim_{n\to\infty} n^{(d-1)/d}/F(n) = 1/c_d$ a.s. By Lemma 6 and the fact that $\sigma_Y^2 < \infty$, we have that $\lim_{n\to\infty} G(n)/n^\delta = 0$ a.s. for any $\delta \geqslant \frac{1}{2}$; we let $\delta = (d-1)/d$ for any $d \geqslant 2$. Multiplying the two latter limit results, we attain $\lim_{n\to\infty} G(n)/F(n) = 0$ a.s., which proves the theorem. □

Remark 2. If $k > 1$ and we had choosen $F(n) = \sum_{j=1}^k L_{\text{TSP}}^j$, it would no longer have been neccesarily true that $Z^*(n,\infty) \geqslant F(n)$, and our proof technique fails. Also, unfortunately, we were not able to attain a similar result for GMP or PAH, even for $k = 1$, as choosing $F(n) = \frac{3}{2} L_{\text{TSP}}$ does not necessarily satisfy $Z^*(n,\infty) \geqslant F(n)$.

4.2.2. Renewal Process Release Dates for Arbitrary k. We begin by stating and proving the main result of this subsection.

Theorem 13. *Under Assumptions* 2 *and* 4, *if* $0 < \mu_X < \infty$, *then* $\lim_{n\to\infty} Z^{\text{GM}}(n,\infty)/Z^*(n,\infty) = 1$ *a.s.*

Proof. To prove this result, we again (c.f. the proof of Theorem 12) first find appropriate random variables $F(n)$ and $G(n)$: Lemma 4 lets us assign $F(n) = R_n$ and $G(n) = \frac{3}{2}\sum_{j=1}^k L_{\text{TSP}}^j$.

By Theorem 11, we have that, in dimension d, there exist positive constants $c_d^j > 0$ such that $\lim_{n\to\infty} L_{\text{TSP}}^j / n^{(d-1)/d} = c_d^j$ a.s. Thus,

$$\lim_{n\to\infty} \frac{G(n)}{n^{(d-1)/d}} = c_d \tag{2}$$

a.s., where $c_d = \frac{3}{2}\sum_{j=1}^k c_d^j$. By the Strong Law of Large Numbers, we have that

$$\lim_{n\to\infty} \frac{F(n)}{n} = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^n X_i = \mu_X \quad \text{a.s.}$$

Because $\mu_X > 0$, we see that

$$\lim_{n\to\infty} \frac{n}{F(n)} = \frac{1}{\mu_X} \tag{3}$$

a.s. and, multiplying Equations (2) and (3), we see that $\lim_{n\to\infty} n^{1/d} G(n)/F(n) = c_d/\mu_X$ a.s. Finally, because $\lim_{n\to\infty} 1/n^{1/d} = 0$, we conclude that $\lim_{n\to\infty} G(n)/F(n) = 0$ a.s. □

By substituting $G(n) = 2\sum_{j=1}^k L_{\text{TSP}}^j$ in the proof of Theorem 13, we have the following corollary for the polynomial-time algorithm GMP.

Corollary 5. *Under Assumptions* 2 *and* 4, *if* $0 < \mu_X < \infty$, *then* $\lim_{n\to\infty} Z^{\text{GMP}}(n,\infty)/Z^*(n,\infty) = 1$ *a.s.*

Finally, by substituting $G(n) = 2\rho L_{\text{TSP}}$ in the proof of Theorem 13, we have the following corollary for the polynomial-time algorithm PAH (for the online TSP, where $k = 1$).

Corollary 6. *Under Assumptions* 2 *and* 4, *if* $0 < \mu_X < \infty$, *then* $\lim_{n\to\infty} Z^{\text{PAH}}(n,\infty)/Z^*(n,\infty) = 1$ *a.s.*

4.3. Capacity Constraints for the Case $k = 1$ in Two-Dimensional Euclidean Space

We return to the special case where each request is simple: $k = 1$. We now consider capacitated online routing problems. The single server has capacity Q. We next define an online algorithm for this problem: Greedy-Capacitated-Routing (GCR).

Algorithm 8 (GCR)

Whenever a new request is released, immediately return to the origin, calculate an optimal set of tours to visit the remaining unserved requests, and begin traversing the tours (in arbitrary order) at unit speed.

Lemma 7.

$$Z^*(n,Q) \geqslant \max\{R_n, Z^{r=0}(n,Q)\} \quad \textit{and}$$

$$Z^{\text{GCR}}(n,Q) \leqslant \min\left\{2R_n + Z^{r=0}(n,Q),\, R_n + \tfrac{1}{2}L_{\text{TSP}} + Z^{r=0}(n,Q)\right\}.$$

Proof. Similar to the proof of Lemma 4. □

Next, we consider both release date structures and give conditions under each where we can show the almost sure asymptotic optimality of GCR. We limit our discussion to requests being located in the two-dimensional Euclidean plane.

4.3.1. Order Statistic Release Dates for $k = 1$. We have two main results for this subsection. The first result is the following.

Theorem 14. *Under Assumptions* 2 *and* 3,

- *If* $d_i = 1\ \forall\, i$, *Q is constant*, $\mu_Y < \infty$, *and* $\mathbb{E}\{d(o,L)\} > 0$, *then* $\lim_{n\to\infty} Z^{\text{GCR}}(n,Q)/Z^*(n,Q) = 1$ *a.s.*
- *If* $d_i = 1\ \forall\, i$, $\lim_{n\to\infty}(Q/\sqrt{n}) = \infty$, *and* $\sigma_Y^2 < \infty$, *then* $\lim_{n\to\infty} Z^{\text{GCR}}(n,Q)/Z^*(n,Q) = 1$ *a.s.*

This theorem is proved in part by using the following result by Haimovich and Rinnooy Kan (1985).

Theorem 15 (Haimovich and Rinnooy Kan 1985). *Under Assumption* 2,

- *If* $d_i = 1\ \forall\, i$ *and Q is constant, then*

$$\lim_{n\to\infty} \frac{Z^{r=0}(n,Q)}{n} = \frac{2\mathbb{E}\{d(o,L)\}}{Q} \quad \textit{a.s.}$$

- *If* $d_i = 1\ \forall\, i$ *and* $\lim_{n\to\infty} Q/\sqrt{n} = \infty$, *then there exists a constant* $\alpha > 0$, *where* $\lim_{n\to\infty} Z^{r=0}(n,Q)/\sqrt{n} = \alpha$ *a.s.*

Proof of Theorem 14. We prove the first part of the theorem first. We take $F(n) = Z^{r=0}(n,Q)$, $G(n) = 2R_n$ and show that $\lim_{n\to\infty} G(n)/F(n) = 0$ a.s. We first decompose the argument of the limit:

$$\frac{G(n)}{F(n)} = 2\left(\frac{n}{Z^{r=0}(n,Q)}\right)\left(\frac{Y_{(n)}}{n}\right).$$

Using the first part of Theorem 15, we conclude that $n/Z^{r=0}(n,Q) \to Q/2\mathbb{E}\{d(o,L)\}$ a.s. Because $\mu_Y < \infty$, Lemma 6 shows that $Y_{(n)}/n \to 0$ a.s. We now prove the second part of the theorem and use the same random variables for $F(n)$ and $G(n)$. Again, we decompose the argument of the limit:

$$\frac{G(n)}{F(n)} = 2\left(\frac{\sqrt{n}}{Z^{r=0}(n, Q)}\right)\left(\frac{Y_{(n)}}{\sqrt{n}}\right).$$

Using the second part of Theorem 15, we conclude that $\sqrt{n}/Z^{r=0}(n, Q) \to (1/\alpha)$ a.s. Because $\sigma_Y^2 < \infty$, Lemma 6 shows that $Y_{(n)}/\sqrt{n} \to 0$ a.s. $\square$

Next, we consider a more general version of the online capacitated routing problem. We begin by normalizing all capacities and demands so that $Q = 1$ and $d_i \leqslant 1\ \forall i$. We now allow the demands d_i to be random variables, i.i.d. from a distribution on $[0, 1]$. Under these conditions, the following result was proved by Bramel et al. (1992), and it proves useful in showing another asymptotic optimality result.

Theorem 16 (Bramel et al. 1992). *Under Assumptions* 2 *and* 5, *there exists a constant* $\phi > 0$ *such that* $\lim_{n\to\infty} Z^{r=0}(n, Q)/n = 2\phi\mathbb{E}\{d(o, L)\}$ *a.s.*

The second result of this subsection is the following.

Theorem 17. *Under Assumptions* 2, 3, *and* 5, *if* $\mu_Y < \infty$ *and* $\mathbb{E}\{d(o, L)\} > 0$, *then* $\lim_{n\to\infty} Z^{\text{GCR}}(n, Q)/Z^*(n, Q) = 1$ *a.s.*

Proof. We take $F(n) = Z^{r=0}(n, Q)$, $G(n) = 2R_n$ and prove that $\lim_{n\to\infty} G(n)/F(n) = 0$ a.s. We rewrite $G(n)/F(n) = 2(n/Z^{r=0}(n, Q))(Y_{(n)}/n)$. Using Theorem 16, we conclude that $n/Z^{r=0}(n, Q) \to (2\phi\mathbb{E} \cdot \{d(o, L)\})^{-1}$ a.s. Because $\mu_Y < \infty$, Lemma 6 tells us that $Y_{(n)}/n \to 0$ a.s. $\square$

4.3.2. Renewal Process Release Dates for $k = 1$. Our main result for this subsection is the following.

Theorem 18. *Under Assumptions* 2 *and* 4, *if* $d_i = 1\ \forall i$, $\lim_{n\to\infty} Q/\sqrt{n} = \infty$, *and* $0 < \mu_X < \infty$, *then*

$$\lim_{n\to\infty} Z^{\text{GCR}}(n, Q)/Z^*(n, Q) = 1 \quad \textit{a.s.}$$

Proof. We take $F(n) = R_n$, $G(n) = \frac{1}{2}L_{\text{TSP}} + Z^{r=0}(n, Q)$ and prove that $\lim_{n\to\infty} G(n)/F(n) = 0$ a.s. We first decompose the argument of the limit:

$$\frac{G(n)}{F(n)} = \left(\frac{n}{\sum_{i=1}^n X_i}\right)\left(\frac{1}{2}\frac{L_{\text{TSP}}}{\sqrt{n}} + \frac{Z^{r=0}(n, Q)}{\sqrt{n}}\right)\left(\frac{1}{\sqrt{n}}\right).$$

By the Strong Law of Large Numbers, we have that $\sum_{i=1}^n X_i/n \to \mu_X$ a.s. and because $\mu_X > 0$, $n/\sum_{i=1}^n X_i \to 1/\mu_X$ a.s. By Theorem 11, we have that there exists $c > 0$ such that $L_{\text{TSP}}/\sqrt{n} \to c$ a.s. By Theorem 15, we have that $Z^{r=0}(n, Q)/\sqrt{n} \to \alpha$ a.s. Finally, because $(c + 2\alpha)/2\mu_X$ is a finite constant and $1/\sqrt{n} \to 0$, the theorem is proved. $\square$

4.4. Online TSP Simulations

In this section, we present simulation results for the online TSP. In this way, we are able to see very precisely the speed of convergence to optimality. Under certain stochastic inputs, the convergence is extremely fast.

Our simulations for the online TSP use algorithm PAH. This algorithm is appealing because its main subroutine calls for solving a classic TSP. For this subroutine, we utilize the powerful Concorde TSP solver by Applegate et al. (2007). Consequently, these results are of a practical interest.

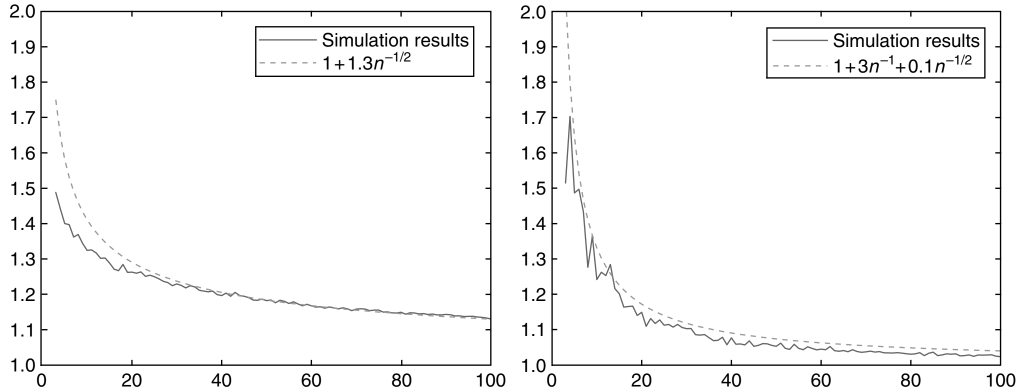
4.4.1. Fast Asymptotic Optimality. We consider the following probabilistic situations. City locations are uniformly distributed on the unit square $[0, 1]^2$. We consider a specific generator for each of the release date structures. We first simulate the case where city release dates are uniformly distributed on $[0, 1]$, and then we simulate the case where the release dates are generated from a Poisson process of parameter 1. For each value of n, we simulate 20 trials and then plot the average ratio of the cost of algorithm PAH to a lower bound on the optimal offline cost: $\rho_n \triangleq Z^{\text{PAH}}(n, \infty)/\max\{R_n, L_{\text{TSP}}\}$; therefore, the plots are conservative. We utilize a lower bound for the offline cost because it is much simpler and efficient than calculating large instances of the TSP with release dates. We also superimpose polynomial functions on the simulation results. These ratios are presented in the left and right plots of Figure 1, respectively.

We now briefly discuss the precision of our simulation results. Clearly, $\rho_n \geqslant 1$. It can be shown that $\rho_n \leqslant 3$. Noting that ρ_n is a random variable, it can be seen that the standard deviation of ρ_n is maximized, equaling one, when $\rho \in \{1, 3\}$, each with probability $\frac{1}{2}$. Our simulation studies estimate the expected value of ρ_n, so the standard error of our estimate, using 20 trials, can be bounded: $\sigma_{\overline{\rho_n}} \triangleq \sigma_{\rho_n}/\sqrt{20} \leqslant 1/\sqrt{20} < 0.23$. Finally, note that these bounds are conservative.

5. Conclusion

The focus of this paper has been on generalizations of the online TSP to allow for precedence constraints, capacity constraints, and multiple vehicles. We derived competitive ratio results for these online problems, several being best-possible. We then considered resource augmentation, where we give the online servers additional resources to offset the powerful offline adversary: faster servers, larger capacities, more servers, and advanced information. We derived improved competitive ratios, several being best-possible. We finally introduced general stochastic structures for the problem data, unknown and unused by the online servers, and showed that our online algorithms are almost surely asymptotically optimal, and we provided computational (Monte Carlo simulations) results showing that these convergences are fast.

Figure 1. Upper bounds on the ratios of the cost of PAH to the optimal offline cost, as a function of n.



Notes. Each data point is the average of 20 trials. The left plot considers release dates that are uniformly distributed on [0, 1]. The right plot considers release dates that are generated from a Poisson process of unit parameter.

We conclude by mentioning two areas of future research. The introduction of accept/reject decisions adds a rich dimension to the problems considered in this paper—the online algorithm has the ability to accept or reject a given request. This introduces additional difficulties; under this framework is it easy to create problem instances with unbounded competitive ratios. Therefore, new measures of online routing algorithms under accept/reject decisions must be designed and utilized. Another area of research is to investigate more fully the value of varying degrees of information about the problem data. For example, if there was a service time at each location, how much would it be worth to know the exact service time, a distribution for the service time, etc? A rich variety of problems are available for investigation.

6. Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at http://or.pubs.informs.org/ecompanion.html.

Acknowledgments

The authors thank the anonymous referees, whose comments and input improved the clarity of the paper and its standing.

References

Allulli, L., G. Ausiello, L. Laura. 2005. On the power of lookahead in on-line vehicle routing problems. *Proc. Eleventh Internat. Comput. Combinatorics Conf., Kunming, Yannan, P. R. China. Lecture Notes in Computer Science*, Vol. 3595. Springer, Berlin/Heidelberg, 728–736.

Altinkemer, K., B. Gavish. 1990. Heuristics for delivery problems with constant error estimates. *Transportation Sci.* **24**(4) 295–297.

Applegate, D., R. Bixby, V. Chvátal, W. Cook. 2007. Concorde software for solving TSP. http://www.tsp.gatech.edu/concorde.html.

Ascheuer, N., S. Krumke, J. Rambau. 2000. Online dial-a-ride problems: Minimizing the completion time. *Proc. 17th Internat. Sympos. Theoret. Aspects Comput. Sci., Lille, France. Lecture Notes in Computer Science*, Vol. 1770. Springer, Berlin/Heidelberg, 639–650.

Ausiello, G., V. Bonifaci, L. Laura. 2005. On-line algorithms for the asymmetric traveling salesman problem. *Proc. 9th Workshop on Algorithms and Data Structures, Waterloo, Canada. Lecture Notes in Computer Science*, Vol. 3608. Springer, Berlin/Heidelberg, 306–317.

Ausiello, G., L. Allulli, V. Bonifaci, L. Laura. 2006. On-line algorithms, real time, the virtue of laziness, and the power of clairvoyance. Working paper, University of Rome "La Sapienza," Rome, Italy.

Ausiello, G., M. Demange, L. Laura, V. Paschos. 2004. Algorithms for the on-line quota traveling salesman problem. *Inform. Processing Lett.* **92**(2) 89–94.

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

Ball, M., M. Queyranne. 2006. Robust revenue management: A competitive analysis of online booking. Robert H. Smith School Research Paper RHS 06-021, University of Maryland, College Park, MD. http://ssrn.com/abstract=896547.

Beardwood, J., J. Halton, J. Hammersley. 1959. The shortest path through many points. *Proc. Cambridge Philos. Soc.* **55** 299–327.

Ben-David, S., A. Borodin. 1994. A new measure for the study of on-line algorithms. *Algorithmica* **11**(1) 73–91.

Ben-Tal, A., A. Nemirovski. 2000. Robust solutions of linear programming problems contaminated with uncertain data. *Math. Programming* **88**(3) 411–424.

Bertsimas, D., M. Sim. 2003. Price of robustness. *Oper. Res.* **52**(1) 35–53.

Blom, M., S. O. Krumke, W. E. de Paepe, L. Stougie. 2001. The online TSP against fair adversaries. *INFORMS J. Comput.* **13**(2) 138–148.

Bonifaci, V., L. Stougie. 2007. Online k-server routing problems. *Proc. 4th Workshop on Approximation and Online Algorithms, Zurich, Switzerland. Lecture Notes in Computer Science*, Vol. 4368. Springer, Berlin/Heidelberg, 83–94.

Bramel, J., E. G. Coffman, P. Shor, D. Simchi-Levi. 1992. Probabilistic analysis of the capacitated vehicle routing problem with unsplit demands. *Oper. Res.* **40**(6) 1095–1106.

Chou, M. C., M. Queyranne, D. Simchi-Levi. 2006a. The asymptotic performance ratio of an on-line algorithm for uniform parallel machine scheduling with release dates. *Math. Programming* **106**(1) 137–157.

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

Feuerstein, E., L. Stougie. 2001. On-line single-server dial-a-ride problems. *Theoret. Comput. Sci.* **268**(1) 91–105.

Frederickson, G., M. Hecht, C. Kim. 1978. Approximation algorithms for some routing problems. *SIAM J. Comput.* **7**(2) 178–193.

Galambos, J. 1987. *The Asymptotic Theory of Extreme Order Statistics*, 2nd ed. Robert E. Krieger Publishing Company, Malabar, FL.

Haimovich, M., A. H. G. Rinnooy Kan. 1985. Bounds and heuristics for capacitated routing problems. *Math. Oper. Res.* **10**(4) 527–542.

Hiller, B. 2005. Probabilistic competitive analysis of a dial-a-ride problem on trees under high load. ZIB Report 05-56, Konrad-Zuse-Zentrum fur Informationstechnik, Berlin, Germany.

Jaillet, P., M. Wagner. 2006. Online routing problems: Value of advanced information as improved competitive ratios. *Transportation Sci.* **40**(2) 200–210.

Kalyanasundaram, B., K. Pruhs. 2000a. The online transportation problem. *SIAM J. Math.* **13**(3) 370–383.

Kalyanasundaram, B., K. Pruhs. 2000b. Speed is as powerful as clairvoyance. *J. Discrete ACM* **47**(4) 617–643.

Koutsoupias, E., C. Papadimitriou. 2000. Beyond competitive analysis. *SIAM J. Comput.* **30**(1) 300–317.

Lan, Y., H. Gao, M. Ball, I. Karaesmen. 2007. Revenue management with limited demand information. Submitted.

Liu, H., M. Queyranne, D. Simchi-Levi. 2005. On the asymptotic optimality of algorithms for the flow shop problem with release dates. *Naval Res. Logist.* **52**(3) 232–242.

Perakis, G., G. Roels. 2006a. The price of information: Inventory management with limited information about demand. *Manufacturing Service Oper. Management* **8**(1) 102–104.

Perakis, G., G. Roels. 2006b. Robust controls for network revenue management. Working paper, Anderson School of Management, University of California, Los Angeles, CA.

Raghavan, P. 1992. A statistical adversary for on-line algorithms. *Online Algorithms, DIMACS Ser. Discrete Math. Comput. Sci.* **7** 79–83.

Sleator, D., R. Tarjan. 1985. Amortized efficiency of list update and paging rules. *Comm. ACM* **28**(2) 202–208.

Young, N. 1994. The k-server dual and loose competitiveness for paging. *Algorithmica* **11**(6) 525–541.