

Almost Sure Asymptotic Optimality for Online Routing and Machine Scheduling Problems

Patrick Jaillet ^{*} and Michael R. Wagner [†]

November 2006, revised May 2007, June 2007

Abstract

In this paper we study algorithms for online routing and machine scheduling problems. The problems are “online” because the problem instances are revealed incrementally. We first study algorithms for the online Traveling Repairman Problem (TRP), where a single server is to visit a set of locations in a network with the objective of minimizing the sum of weighted completion times. We then analyze well-known online algorithms for a variety of machine scheduling problems, which are appropriate models for many network optimization problems; in the scheduling notation of Graham et al. [18], we consider $1|r_j, pmtn|\sum_j w_j C_j$, $1|r_j|\sum_j w_j C_j$, $Q|r_j, pmtn|\sum_j C_j$, $P|r_j|\sum_j C_j$, $Q|r_j, pmtn|\sum_j w_j C_j$ and $Q|r_j|\sum_j w_j C_j$.

We introduce general probabilistic assumptions about the problem data as a *tool* to study the online algorithms for these online combinatorial problems. The algorithms do not utilize the underlying probabilistic assumptions in any way. We prove that these online algorithms are almost surely asymptotically optimal.

1 Introduction

In this paper we consider a number of online combinatorial optimization problems. In particular, we consider an online routing problem and several online machine scheduling problems. These problems are all appropriate models for various network optimization problems; we provide examples shortly. The problems are online because the problem instance is revealed incrementally but decisions can (and sometimes must) be made before the entire problem instance is revealed. We investigate these problems in a novel manner: we introduce general probabilistic assumptions for the problem data and we analyze classic online algorithms that do not utilize the stochastic knowledge. We prove that these well-known online algorithms are asymptotically optimal, almost surely.

^{*}Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, jaillet@mit.edu.

[†]Department of Management, California State University East Bay, Hayward, CA 94542, michael.wagner@csueastbay.edu.

The first problem that we study is the Traveling Repairman Problem (TRP), which is well known in both Operations Research and Computer Science. In one of its simplest forms we are given a network $N = (V, E)$, where vertices represent cities and edge lengths represent distances between cities. Each city has an associated non-negative weight, representing, for example, the importance of the city. A city's completion time is defined as the first time that a city is visited. Given an origin city, the task is to find a path through the network that traverses each city at least once. Assuming that the repairman has a constant speed, the objective is to minimize the weighted sum of city completion times; this objective is also referred to as the *latency*. The latency is closely related to the (weighted) average completion date of all cities. 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 "TRP with release dates." Additionally, we may incorporate 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.

In our paper, we study online versions of the TRP with precedence constraints, where the instance is not revealed all at once. In the framework considered in this paper, the cities are revealed dynamically over time, independent of the repairman's location, at their release date. The corresponding offline problem, where all data is known a priori, is the TRP with release dates and precedence constraints as introduced above.

Apart from the straightforward applications in routing with the latency objective, the TRP has many other applications. Simchi-Levi and Berman [31] consider the TRP in flexible manufacturing systems. Some machine scheduling problems can be cast as a TRP; see Rinnooy Kan [23] and Picard and Queyranne [26]. Tsitsiklis [34] describes other applications as well. Finally, the TRP is appropriate in searching problems: if one were to search for a prize located at any of n given points in a network (where distances satisfy the triangle inequality) with equal probability, the optimal TRP solution gives the minimum expected time to find the prize (see Blum, Chalasani, Coppersmith, Pulleyblank, Raghavan, Sudan [6].)

We also study online algorithms for a number of machine scheduling problems whose offline versions are NP-hard. Machine schedule problems are an appropriate model for solving a number of network optimization problems. Generically, any routing optimization problem on a network can be cast as a machine scheduling problem with sequence-dependent processing times. More specifically, scheduling in computer networks can be solved using machine scheduling models. For example, single machine scheduling concepts have been applied to browsing the Internet by Xia and Tse [35]. Bampis and Rouskas [4] apply machine scheduling concepts to problems arising in optical networks and IP routers.

We study the problem of scheduling jobs that arrive in an online fashion in single and multiple machine environments. In the multiple machine environment, we consider the case where the machines are all identical as well as the case where the machines have different processing speeds. A job's completion time is defined as the first time that the job has been completely processed. The

objective is to minimize the weighted sum of job completion times (similarly to the online TRP objective). We consider both preemptive and non-preemptive problems. In the scheduling notation of Graham, Lawler, Lenstra, Rinnooy Kan [18], we study online versions of $1|r_j, pmtn|\sum_j w_j C_j$, $1|r_j|\sum_j w_j C_j$, $Q|r_j, pmtn|\sum_j C_j$, $P|r_j|\sum_j C_j$, $Q|r_j, pmtn|\sum_j w_j C_j$ and $Q|r_j|\sum_j w_j C_j$.

The focus of this paper is on studying algorithms for these online routing and machine scheduling problems. Online algorithms are usually evaluated using the competitive ratio, which is defined as the worst case ratio, over all problem instances, of the online algorithm's cost to the cost of an optimal offline algorithm. We, however, evaluate these online algorithms using the *asymptotic* 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, *for large enough problem instance sizes*.

We show that under certain conditions, the asymptotic competitive ratio of classic online algorithms is equal to one; i.e., the online algorithms are asymptotically optimal. In particular, we introduce general probabilistic assumptions on the problem data as a *tool* to study the online algorithms. The *deterministic* online algorithms that we study do not use the probabilistic information in any way. Furthermore, no specific distributional assumptions are made; we only assume that the problem data is generated by a distribution that belongs to a class of distributions that we define. Under these stochastic assumptions, we show that the classic algorithms we consider are almost surely asymptotically optimal. One of the motivations for this research is to provide an explanation of the excellent performance that some of these algorithms exhibit computationally as well as in practice.

1.1 Previous Work

Considering the online TRP, Feuerstein and Stougie [15] give a lower bound of $(1 + \sqrt{2})$ for the competitive ratio and a 9-competitive algorithm, both for the online TRP on the real line. Krumke, de Paepe, Poensgen, Stougie [24] improved upon this result to give a $(1 + \sqrt{2})^2$ -competitive deterministic algorithm for the online TRP in general metric spaces as well as a Θ -competitive randomized algorithm, where $\Theta \approx 3.64$; this result was corrected to $\Theta \approx 3.86$ in Jaillet and Wagner [20] (see also [25]). The online TRP under advanced information, where cities are revealed to the online algorithm *before* their release dates, was also considered in [20]. A similar approach was taken by Allulli, Ausiello, Laura [2] in the form of a *lookahead*. Bonifaci and Stougie [8] consider the online TRP with k servers and give an algorithm that is 6-competitive and mention that their approach can be modified to prove their algorithm is $(1 + \sqrt{2})^2$ -competitive, matching the single server result. They also consider the effect on the competitive ratio of giving the online algorithm additional servers: If the online algorithm has k servers and the offline algorithm has $k^* \leq k$ servers, their algorithm is $2 \cdot 3^{1/\lfloor k/k^* \rfloor}$ -competitive.

To the best of our knowledge, there is very little previous research on the *asymptotic* competitive ratio for the online TRP. The one exception is also contained in the work by Bonifaci and Stougie [8]: If all cities are located on the real line, they give a deterministic algorithm with a competitive

ratio of $1 + O((\log k)/k)$; i.e., as $k \rightarrow \infty$, their algorithm is asymptotically optimal.

However, there have been similar approaches for related online routing problems such as the online TSP. Hiller [19] performs a probabilistic asymptotic competitive analysis of an online Dial-a-Ride problem on trees. Recently, Jaillet and Wagner [21] have investigated generalizations of the online TSP from an asymptotic point of view and have shown a number of almost sure asymptotic optimality results.

We next consider the literature on online machine scheduling problems. As the literature in this area is vast and we do not intend to give a comprehensive review, we only mention the references that are most relevant to our paper. Anderson and Potts [3] give a deterministic online algorithm for $1|r_j|\sum_j w_j C_j$ and show that it has a competitive ratio of 2. Goemans, Queyranne, Schulz, Skutella and Wang [17] give a randomized algorithm for the same problem with a competitive ratio of at most 1.68. Sitters [32] gives a deterministic algorithm for $1|r_j, pmtn|\sum_j w_j C_j$ with a competitive ratio of at most 1.56. Schulz and Skutella [29] give a randomized online algorithm for the same problem with a competitive ratio of at most $\frac{4}{3}$. Chekuri, Motwani, Natarajan and Stein [10] give a deterministic online algorithm for $P|r_j|\sum_j C_j$ that has a competitive ratio of at most $3 - \frac{1}{m}$, where m is the number of machines. We are not aware of any further competitive ratio results for the problems $P|r_j|\sum_j C_j$ and $P|r_j, pmtn|\sum_j C_j$; therefore, we now give state of the art results for the *weighted* sum of completion dates objective. Correa and Wagner [13] give a deterministic algorithm for $P|r_j|\sum_j w_j C_j$ with a competitive ratio of at most 2.62. Schulz and Skutella [30] give randomized online algorithms for $P|r_j|\sum_j w_j C_j$ and $P|r_j, pmtn|\sum_j w_j C_j$ that have competitive ratios of at most 2. Correa and Wagner [13] improve on these last results, but the improvement depends on the number of machines. To the best of our knowledge, there are no existing worst-case results for $Q|r_j|\sum_j w_j C_j$ or its preemptive version.

Most relevant to our online machine scheduling work is the paper by Chen and Shen [11]. These authors study online single machine, uniform parallel machine and flow shop scheduling problems under stochastic assumptions on the problem data, similar to ours, and they show that a class of online algorithms are almost surely asymptotically optimal. However, the authors make the additional assumption that there exist explicit positive lower and upper bounds on the job weights and processing requirements; we do not make this assumption. Similarly, Chou, Queyranne and Simchi-Levi [12] consider the online version of $Q|r_j|\sum_j w_j C_j$ and show that, if there exist positive lower and upper bounds on the processing requirements and weights, the algorithm *Weighted Shortest Processing Requirement* (WSPR) is deterministically asymptotically optimal. Additionally, Kaminsky and Simchi-Levi [22] show that the *Shortest Processing Time Among Available* (SPTA) heuristic is deterministically asymptotically optimal for the online version of $1|r_j|\sum_j C_j$, under the assumption that the processing requirements are all contained in a bounded interval.

The paper by Savelsbergh, Uma and Wein [28] gives a comprehensive computational study of the offline $1|r_j|\sum_j w_j C_j$; they show that many algorithms perform much better than is theoretically predicted. In the journal version of Correa and Wagner [13], a computational study of online

algorithms for $P|r_j|\sum_j w_j C_j$ and $P|r_j, pmtn|\sum_j w_j C_j$ is given and it is again shown that the algorithms perform better than expected. Additionally, Albers and Schröder [1] perform a computational study of online algorithms for parallel machine scheduling problems with the objective of minimizing the makespan. They experimentally show that online algorithms that perform well on randomly generated data do not necessarily perform well on real-world data.

1.2 Our Contributions

We give the first asymptotic optimality results for the online TRP with precedence constraints. These results continue the work in [21] to arguably more complicated online routing problems (routing optimization problems with the latency objective are usually considered more difficult than those with the makespan objective). Furthermore, our asymptotic approach is arguably more realistic than the only other asymptotic analysis for the online TRP (see [8]): We consider the limit as the number of cities (rather than the number of servers) goes to infinity. In our opinion, it is more realistic to consider a problem with a very large number of locations to visit than one with a very large fleet of vehicles. These results also have a strong practical implication: If the number of locations to visit is large enough, then the additional cost of having dynamic uncertainty (the problem being online), compared to having all information a priori, is negligible.

We also give the first asymptotic optimality results for the problems $1|r_j, pmtn|\sum_j w_j C_j$, $1|r_j|\sum_j w_j C_j$, $Q|r_j, pmtn|\sum_j C_j$, $P|r_j|\sum_j C_j$, $Q|r_j, pmtn|\sum_j w_j C_j$ and $Q|r_j|\sum_j w_j C_j$, that do not require explicit bounds on the job weights and processing requirements. For the first three problems, we analyze well-known online algorithms that run in polynomial time. For the final two problems, we study online algorithms that do not run in polynomial time. Our results complement the research in [11], [12] and [22]. Taken together, these results provide a convincing explanation for the good practical and computational performance exhibited by the online algorithms.

A main benefit of our machine scheduling research is to relax the explicit lower and upper bounds required by the deterministic analyses in [12] and [22]. By introducing probabilistic assumptions, we are able to relax the explicit bounds on the data into bounds on the distributional moments of the data. Note, however, that our approach was only successful for slightly simpler problems than those considered in [12]. Additionally, our proofs, while in a probabilistic domain, are arguably simpler. Finally, our results are similar to the almost sure asymptotic optimality results in [11], but again we do not require the explicit bounds on the problem data.

Paper Outline: In Section 2, we detail the problem descriptions, introduce probabilistic assumptions and give useful technical results. In Section 3 we present our results for the online TRP. In Section 4 we consider single machine scheduling problems and in Section 5 we study the multiple machine cases.

2 Preliminaries

2.1 Problem Data for the Online Traveling Repairman Problem

The data for our problem is given by (\mathbf{l}_i, r_i, w_i) , $i = 1, \dots, n$, where n is the number of requests. Each request i consists of m locations to visit: $\mathbf{l}_i = (l_i^1, l_i^2, \dots, l_i^m)$, where $l_i^j \in \mathcal{M}$, \mathcal{M} an Euclidean metric space of dimension d . The quantity $r_i \in \mathbb{R}_+$ is the i^{th} request's release date; i.e., r_i is the time after which cities in request i will accept service. We assume, without loss of generality, that $r_1 \leq r_2 \leq \dots \leq r_n$. The quantity $w_i \in \mathbb{R}_+$ is the i^{th} request's weight. Strict precedence constraints exist *within* a request: $\forall i, l_i^1 \preceq l_i^2 \preceq \dots \preceq l_i^m$. 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 $\{l_1^j, \dots, l_n^j\}$ for each $j \in \{1, \dots, m\}$. The service requirement at a city is zero. Unless stated otherwise, the repairman travels at unit speed or is idle. The problem begins at time 0, and the repairman is initially at a designated origin o of the metric space. The objective is to minimize the weighted sum of *request* completion times $\sum_{i=1}^n w_i C_i$, where a request's completion time is the first time that all cities in the request have been visited. Finally, L_{TRP} is the optimal cost when all release dates are equal to zero.

2.2 Problem Data for Online Machine Scheduling Problems

The data for our problems is given by (p_i, r_i, w_i) , $i = 1, \dots, n$, where n is the number of jobs. $p_i \in \mathbb{R}_+$ is the processing requirement of job i , $r_i \in \mathbb{R}_+$ is the release date of job i and $w_i \in \mathbb{R}_+$ is the weight associated with job i ; the problems are online because job i 's existence and data do not become known until time r_i . We consider scheduling jobs on (1) a single machine, (2) parallel identical machines and (3) parallel uniform machines (i.e., machines have different job processing speeds). In the multiple machine environment, we have $m \in \mathbb{Z}$ machines available, where m is fixed. The objective is to schedule the jobs on the machine(s) to minimize the weighted sum of completion dates $\sum_j w_j C_j$. We consider both preemptive and non-preemptive problems; when preemption is allowed, a job can be interrupted and resumed later, possibly on a different machine. We study online versions of $1|r_j, pmtn|\sum_j w_j C_j$, $1|r_j|\sum_j w_j C_j$, $Q|r_j, pmtn|\sum_j C_j$, $P|r_j|\sum_j C_j$, $Q|r_j, pmtn|\sum_j w_j C_j$ and $Q|r_j|\sum_j w_j C_j$. Let $s_1 \geq s_2 \geq \dots \geq s_m > 0$ be the speeds of the machines in the Q case; in the P case, $s_j = 1, \forall j$. Job i on machine j will take p_i/s_j time to complete.

2.3 Online Optimization and Competitive Analysis

From the online perspective, the total number of requests/jobs, represented by the parameter n , is not known, and request/job i only becomes known at time r_i . Let Z_n^A denote the cost of online algorithm A on an instance of n cities and Z_n^* is the corresponding optimal offline cost where all data is known a priori. The problem instance underlying Z_n^A and Z_n^* will be clear from context.

The performance of online algorithms is usually measured using the competitive ratio and the

asymptotic competitive ratio criteria. The competitive ratio is defined as the worst-case ratio, over all problem instances, of online to offline costs: $\max_{instances} Z_n^A/Z_n^*$. 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 $n \geq n_0$, $Z_n^A/Z_n^* \leq c$. An online algorithm is said to be best-possible if there does not exist another online algorithm with a strictly smaller competitive ratio.

2.4 Stochastic Assumptions for the Online Traveling Repairman Problem

We now list two different stochastic assumptions for the online TRP that are called upon throughout this paper. We use uppercase letters to denote random variables.

Assumption 1 (Locations) *For each $j \in \{1, \dots, m\}$, $L_1^j, L_2^j, \dots, L_n^j$ are independently identically distributed from a distribution of compact support in $d \geq 2$ 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 $k = l$).*

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

Assumption 2 (Release Dates) *The release date of each request is a realization of a generic non-negative random variable $Y \geq 0$; i.e., the unordered release dates are independently identically distributed from a given distribution. As our model requires an order ($R_k \leq R_l$ for $k < l$), the k -th release date is the k -th order statistic: $R_k = Y_{(k)}$, where $Y_k \geq 0$, $k = 1, \dots, n$ are i.i.d. random variables and $Y_{(1)} \leq Y_{(2)} \leq \dots \leq Y_{(n)}$.*

We also consider a renewal process structure for the release dates in Section 3.3; since Section 3.3 is the only place in our paper where we apply this alternate structure, we define the assumption in that section. We also utilize a deterministic assumption on the city weights, which we detail next.

Assumption 3 (Weights) *There exist values $0 < \omega \leq \Omega$ such that $\omega \leq w_i \leq \Omega$, $\forall i$.*

The lower bound of ω in Assumption 3 simply eliminates requests with zero weight, requests which would not have been counted in the objective function cost anyway. The upper bound of Ω is intended to eliminate the pathological case where a single request has an arbitrary large weight which dominates the objective function cost.

2.5 Stochastic Assumptions for the Online Machine Scheduling Problems

We now list the different stochastic assumptions for the online machine scheduling problems.

Assumption 4 (Release Dates) *The job release dates satisfy Assumption 2.*

Assumption 5 (Processing Requirement) *The processing requirement P_i of each job is a realization of a generic non-negative random variable $P \geq 0$; i.e., the processing requirements are independently identically distributed from a given distribution.*

Assumption 6 (Weights) *The weight W_i of each job is a realization of a generic non-negative random variable $W \geq 0$; i.e., the weights are independently identically distributed from a given distribution.*

2.6 Discussion of Stochastic Assumptions

The appeal of our stochastic assumptions is that they do not specify any particular distribution for the data. The assumptions only introduce a probabilistic structure for the data. Furthermore, these structures match many of the assumptions made in computational studies that appear in the literature (e.g., see [1], [13], [28]); therefore, our theoretical analysis complements the computational studies. However, our assumptions have limitations. Our model requires that related data (such as job processing requirements) are identically independently distributed.

The assumption of independence precludes the application of our analysis to any practical setting where there are strong correlations between requests. For example, suppose the online TRP model is applied to an ice-cream truck that travels in neighborhoods to sell ice cream to children, who are the requests ($m = 1$). Children see that other children are buying ice cream from the truck and many will also want ice cream. Therefore, the requests in this case are highly correlated and our model is not appropriate.

The assumption that certain data are identically distributed is less of a concern. Even if the characteristics of individual requests are generated from different distributions, our analysis can be extended by utilizing generalized versions of our analysis tools (e.g., Kolmogorov’s Strong Law of Large Numbers). However, this assumption allows us to bypass the additional technical details and the main ideas of our proofs are more easily accessible to the reader.

2.7 Technical Details

In this subsection, we present useful technical results.

Theorem 1 (Beardwood, Halton, Hammersley [5]) *Under Assumption 1, there exists a $c_d^j > 0$ such that $\lim_{n \rightarrow \infty} \frac{L_{TRP}^j}{n^{(d-1)/d}} = c_d^j$ almost surely, where d is the dimension of the underlying Euclidean space.*

Lemma 1 (Bompadre, Dror, Orlin [7]) *The cost L_{TRP} of the minimum latency problem with unit weights when n cities are uniformly distributed in $[0, 1]^2$ is $\Omega(n^{3/2})$ almost surely.*

Lemma 2 *Let $\{X_i\}$ be a sequence of non-negative i.i.d. random variables. If $\mathbb{E}[X^r] < \infty$, then*

$$\lim_{n \rightarrow \infty} \frac{\max_{1 \leq i \leq n} X_i}{n^\delta} = 0,$$

almost surely, for all $\delta \geq \frac{1}{r}$.

Proof Consequence of Theorem 4.4.1 in Galambos [16]. ■

Lemma 3 Let $\{X_i\}$ be a sequence of i.i.d. random variables. If $\mathbb{E}[X^2] < \infty$, then

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n jX_j}{n^2} = \frac{\mathbb{E}[X]}{2},$$

almost surely.

Proof The martingale M_n defined by $M_n = \sum_{j=1}^n \frac{X_j - \mathbb{E}[X_j]}{j}$ is bounded in L^2 , so is convergent almost surely. Using Kronecker's lemma we then conclude that

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n j(X_j - \mathbb{E}[X_j])}{n^2} = 0,$$

almost surely. ■

3 The Online TRP with Precedence Constraints

We consider here the general case $m \geq 1$. Note that when $m = 1$ we have the classic online Traveling Repairman Problem and when $m = 2$, we have an online version of the latency-objective Dial-a-Ride problem.

We use a generic technique to prove almost sure asymptotic optimality: We find random variables F_n and G_n that satisfy $Z_n^* \geq F_n$ and $Z_n^A \leq F_n + G_n$ for all n for some online algorithm A . Then, we show that $\lim_{n \rightarrow \infty} G_n/F_n = 0$, almost surely, which implies that $\lim_{n \rightarrow \infty} Z_n^A/Z_n^* = 1$, almost surely.

This section is organized as follows: In Section 3.1 we give online algorithms and derive upper bounds on their costs as well as lower bounds on the optimal offline costs. In Section 3.2, we prove almost sure asymptotic optimality results for the case where the release dates satisfy Assumption 2. Finally, we prove similar results in Section 3.3 when the release dates are instead generated by a general renewal process.

3.1 Algorithms and Bounds

We define the strategy Greedy-Latency (GL) for these problems, followed by two polynomial-time strategies.

Algorithm 1 : GL At any release date, calculate a path \mathcal{P} of minimum latency that satisfies the following constraints:

1. \mathcal{P} starts at the current server location.
2. All unserved requests are visited and the precedence constraints are respected.

3. If there are no unserved requests, remain idle at the current location (not necessarily the origin).

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

We next define the polynomial-time strategy Greedy-Latency-Polynomial (GLP) for the special case where $m = 1$ and $w_i = 1, \forall i$.

Algorithm 2 : GLP At any release date, use a ρ -approximation algorithm for minimizing latency to find a path \mathcal{P} beginning at the current server location and visiting all unserved requests. Then the server traverses \mathcal{P} at unit speed, until the next release date (if any). If there are no unserved locations, remain idle at the current location (not necessarily the origin).

To the best of our knowledge, there are no approximation algorithms for the arbitrary weight case. Also to the best of our knowledge, the approximation algorithm (for the unit weight case) with the smallest approximation ratio ρ to date is the one given by Chaudhuri, Godfrey, Rao and Talwar [9], which has $\rho < 3.6$.

Finally, we give a simple polynomial-time algorithm for the general case: Serve-In-Order-Received (SIOR).

Algorithm 3 : SIOR Serve in the order received; i.e., visit the locations in the order:

$$L_1^1, \dots, L_1^m, L_2^1, \dots, L_2^m, \dots, L_n^1, \dots, L_n^m.$$

When there are no known unserved locations, remain idle at the current location.

We now derive useful bounds for the costs of these algorithms, as well as for the optimal offline cost, in a series of lemmas and corollaries. We consider separately the cases $m = 1$ and an arbitrary value of m . We first consider the case where $m = 1$.

Lemma 4 If $m = 1$, $Z_n^* \geq L_{TRP}$ and $Z_n^{GL} \leq 2R_n \sum_{i=1}^n w_i + L_{TRP}$.

Proof The lower bound on Z_n^* is clear. Now we consider the server (repairman) at time R_n . Consider an alternate strategy where the server returns to the origin and then serves *all* cities optimally; this strategy clearly has a larger latency than GL since GL does not necessarily return to the origin at time R_n and may have already served some cities. The initial return to the origin of this alternate strategy takes at most R_n time since the server moves at unit speed. The (alternate) server then proceeds on the optimal path that minimizes the latency through all n cities. The completion time of request i in the alternate strategy is $2R_n + C_i^*$, which implies that the cost of GL is at most $2R_n \sum_{i=1}^n w_i + L_{TRP}$. ■

The following corollary is immediate.

Corollary 1 If $m = 1$ and $w_i = 1, \forall i$, $Z_n^{GLP} \leq 2nR_n + \rho L_{TRP}$.

We now consider the situation where m is arbitrary.

Lemma 5

$$Z_n^* \geq \sum_{j=1}^n w_j R_j \quad \text{and} \quad Z_n^{GL} \leq \sum_{j=1}^n w_j \left(R_j + 3 \sum_{i=1}^m L_{TSP}^i \right).$$

Proof We begin with the lower bound on Z_n^* . Clearly, the optimal completion time of each request is at least its release date; thus we have $Z_n^* \geq \sum_{j=1}^n w_j R_j$.

We now show the upper bound on Z_n^{GL} by induction on the number of requests n . For $n = 1$ (subscripts are suppressed), with $L^0 = o$, it is clear that

$$\begin{aligned} Z^{GL} &= w \left(R + \sum_{i=1}^m d(L^{i-1}, L^i) \right) \\ &\leq w \left(R + \sum_{i=1}^m (d(L^{i-1}, o) + d(o, L^i)) \right) \\ &\leq w \left(R + \sum_{i=1}^m 2d(o, L^i) \right) \\ &= w \left(R + \sum_{i=1}^m L_{TSP}^i \right). \end{aligned}$$

Now, assuming $Z_{n-1}^{GL} \leq \sum_{j=1}^{n-1} w_j (R_j + 3 \sum_{i=1}^m L_{TSP}^i(n-1))$, $L_{TSP}^i(n-1)$ being the shortest tour through the locations L_1^i, \dots, L_{n-1}^i , and noting that $L_{TSP}^i(n-1) \leq L_{TSP}^i(n) \triangleq L_{TSP}^i$, we shall prove the result for n . Define C_{n-1}^{max} as the (projected) maximum completion time of all requests in the instance of $(n-1)$ requests. We first find an upper bound on C_{n-1}^{max} . Recall that GL performed a re-optimization at time R_{n-1} . Consider an alternate server that, at time R_{n-1} , first returned to the origin before proceeding to visit all unserved requests; this return takes at most

$$\max_{1 \leq i \leq m} \left\{ \max_{1 \leq j \leq n-1} d(o, L_j^i) \right\} \leq \max_{1 \leq i \leq m} \left\{ \frac{1}{2} L_{TSP}^i \right\} \leq \frac{1}{2} \sum_{i=1}^m L_{TSP}^i$$

time. Once the alternate server reaches the origin, it first travels through the locations $\{L_1^1, \dots, L_{n-1}^1\}$, then $\{L_1^2, \dots, L_{n-1}^2\}$ and so on until $\{L_1^m, \dots, L_{n-1}^m\}$. This takes at most $\sum_{i=1}^m L_{TSP}^i$ time. Since C_{n-1}^{max} for GL is clearly at most the respective value for this alternate strategy, we have that

$$C_{n-1}^{max} \leq R_{n-1} + \frac{3}{2} \sum_{i=1}^m L_{TSP}^i \leq R_n + \frac{3}{2} \sum_{i=1}^m L_{TSP}^i.$$

Re-optimizing at time R_n will result in a latency value that is no more than that of the following strategy: Wait until requests $1, \dots, (n-1)$ have all been served and then serve request n . Letting \tilde{C}_n denote the completion time of request n in this virtual strategy and noting that at time C_{n-1}^{max}

the server is at a location L_j^m , $j \in \{1, \dots, n-1\}$, we have that

$$\begin{aligned}
Z_n^{GL} &\leq Z_{n-1}^{GL} + w_n \tilde{C}_n \\
&= Z_{n-1}^{GL} + w_n \left(C_{n-1}^{max} + d(L_j^m, L_n^1) + \sum_{i=2}^m d(L_n^{i-1}, L_n^i) \right) \\
&\leq Z_{n-1}^{GL} + w_n \left(C_{n-1}^{max} + d(L_j^m, o) + d(o, L_n^1) + \sum_{i=2}^m (d(L_n^{i-1}, o) + d(o, L_n^i)) \right) \\
&\leq Z_{n-1}^{GL} + w_n \left(C_{n-1}^{max} + \frac{1}{2} L_{TSP}^m + \sum_{i=1}^m 2d(o, L_n^i) \right) \\
&\leq Z_{n-1}^{GL} + w_n \left(C_{n-1}^{max} + \frac{1}{2} L_{TSP}^m + \sum_{i=1}^m L_{TSP}^i \right) \\
&\leq Z_{n-1}^{GL} + w_n \left(C_{n-1}^{max} + \frac{3}{2} \sum_{i=1}^m L_{TSP}^i \right) \\
&\leq Z_{n-1}^{GL} + w_n \left(R_n + 3 \sum_{i=1}^m L_{TSP}^i \right);
\end{aligned}$$

applying the inductive hypothesis proves the lemma. ■

The proof of Lemma 5 also directly applies to strategy *SIOR*:

Corollary 2

$$Z_n^{SIOR} \leq \sum_{j=1}^n w_j \left(R_j + 3 \sum_{i=1}^m L_{TSP}^i \right).$$

3.2 Order Statistic Release Dates for the case $m = 1$ and $w_i = 1, \forall i$

Our main result for this subsection is the following.

Theorem 2 *Under Assumption 2, if $m = 1$, $w_i = 1, \forall i$, $\mathbb{E}[Y^3] < \infty$ and L_1, \dots, L_n are uniformly distributed in $[0, 1]^2$, then*

$$\lim_{n \rightarrow \infty} \frac{Z_n^{GL}}{Z_n^*} = 1$$

almost surely.

Proof We first find appropriate random variables F_n and G_n . By Lemma 4 we let $F_n = L_{TRP}$ and $G_n = 2nR_n$. By Lemma 1, we have that $L_{TRP} = \Omega(n^{3/2})$ almost surely. Since L_{TRP} is almost surely positive, we may conclude that $\frac{1}{L_{TRP}} = O(\frac{1}{n^{3/2}})$ almost surely. For any $\epsilon > 0$, we have that $\frac{1}{L_{TRP}} = o(\frac{n^\epsilon}{n^{3/2}})$ almost surely. Equivalently, we have that $\lim_{n \rightarrow \infty} \frac{n^\gamma}{L_{TRP}} = 0$ almost surely, for any $\gamma < \frac{3}{2}$. Next, we decompose the limit:

$$\frac{G_n}{F_n} = \frac{2nY_{(n)}}{L_{TRP}} = 2 \frac{n^{4/3}}{L_{TRP}} \frac{Y_{(n)}}{n^{1/3}}.$$

Taking limits, with $\gamma = \frac{4}{3}$ and applying Lemma 2 (with $r = 3$ and $\delta = \frac{1}{3}$), proves the theorem.

■

Remark 1 *We actually only require that there exists $\varepsilon > 0$ such that $\mathbb{E}[Y^{2+\varepsilon}] < \infty$ to prove the above theorem.*

Unfortunately, we were unable to prove a similar asymptotic optimality result for GLP. Corollary 1 and the proof of Theorem 2 suggest choosing $F_n = \rho L_{TRP}$. But since $\rho > 1$, it would have no longer been necessarily true that $Z_n^* \geq F_n$. However, the same approach does yield the following corollary.

Corollary 3 *Under Assumption 2, if $m = 1$, $w_i = 1$, $\forall i$, $\mathbb{E}[Y^3] < \infty$ and L_1, \dots, L_n are uniformly distributed in $[0, 1]^2$, then*

$$\lim_{n \rightarrow \infty} \frac{Z_n^{GLP}}{Z_n^*} = \rho$$

almost surely.

3.3 Renewal Process Release Dates for arbitrary m and w_i

We first introduce a new stochastic assumption for the release dates.

Assumption 7 (Renewal Process) *Define non-negative i.i.d. random variables $X_i \geq 0$ to be the time between the $(i-1)^{th}$ and i^{th} 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 .*

The main result of this subsection is the following.

Theorem 3 *Under Assumptions 1, 7 and 3, if $\mathbb{E}[X^2] < \infty$, then*

$$\lim_{n \rightarrow \infty} \frac{Z_n^{GL}}{Z_n^*} = 1$$

almost surely.

Proof We assume $\mu_X > 0$ without loss of generality since otherwise all release dates would be zero, almost surely, and there would be nothing to prove.

We assign $F_n = \sum_{j=1}^n w_j R_j$ and $G_n = 3 \left(\sum_{j=1}^n w_j \right) \left(\sum_{i=1}^m L_{TSP}^i \right)$, in accordance with Lemma 5. Showing $\lim_{n \rightarrow \infty} \frac{G_n}{F_n} = 0$ almost surely proves the theorem. We first bound (using Assumption 3) the argument of the limit:

$$\frac{G_n}{F_n} = \frac{3 \left(\sum_{j=1}^n w_j \right) \left(\sum_{i=1}^m L_{TSP}^i \right)}{\sum_{j=1}^n w_j R_j} \leq \frac{3n\Omega \sum_{i=1}^m L_{TSP}^i}{\omega \sum_{j=1}^n R_j}.$$

We now express the sum of release dates in terms of the X random variables:

$$\sum_{i=1}^n R_i = \sum_{i=1}^n \sum_{j=1}^i X_j = \sum_{j=1}^n \sum_{i=j}^n X_j = \sum_{j=1}^n (n-j+1)X_j = \sum_{j=1}^n jX_j,$$

where the last equality follows (almost surely) from the fact that the X_j random variables are i.i.d.

Next, we take limits and apply Lemma 3 and Theorem 1:

$$\begin{aligned} \frac{3n\Omega \sum_{i=1}^m L_{TSP}^i}{\omega \sum_{j=1}^n R_j} &= \frac{3n\Omega \sum_{i=1}^m L_{TSP}^i}{\omega \sum_{j=1}^n jX_j} \\ &= \left(\frac{3\Omega}{\omega} \right) \left(\frac{n^2}{\sum_{j=1}^n jX_j} \right) \left(\frac{\sum_{i=1}^m L_{TSP}^i}{n^{(d-1)/d}} \right) \left(\frac{1}{n^{1/d}} \right) \\ &\rightarrow \left(\frac{3\Omega}{\omega} \right) \left(\frac{2}{\mu_X} \right) \left(\sum_{i=1}^m c_d^i \right) (0), \end{aligned}$$

almost surely, and the convergence is proved. ■

Since the upper bound on the cost of SIOR is identical to that of GL (c.f. Lemma 5 and Corollary 2), we have the following corollary for the polynomial-time SIOR.

Corollary 4 *Under Assumptions 1, 7 and 3, if $\mathbb{E}[X^2] < \infty$, then*

$$\lim_{n \rightarrow \infty} \frac{Z_n^{SIOR}}{Z_n^*} = 1$$

almost surely.

4 Single Machine Minsum Online Scheduling

We consider online versions of the single machine scheduling problems $1|r_j, pmtn|\sum_j w_j C_j$ and $1|r_j|\sum_j w_j C_j$; offline versions of both these problems are NP-hard.

4.1 Online $1|r_j, pmtn|\sum_j w_j C_j$

Consider the preemptive *Weighted Shortest Processing Requirement* (WSPR) heuristic, which is an online algorithm: At any point in time, among the known unfinished jobs, process the one with the highest ratio w_i/p_i . Note that the WSPR heuristic (also known as Smith's ratio rule [33]) solves $1|\sum_j w_j C_j$, and consequently $1|pmtn|\sum_j w_j C_j$, exactly; e.g., see Pinedo [27]. We begin by stating the main result for this subsection.

Theorem 4 *Under Assumptions 4, 5 and 6, if $\mathbb{E}[Y] < \infty$, then the WSPR heuristic is almost surely asymptotically optimal for the online version of $1|r_j, pmtn|\sum_j w_j C_j$.*

Proof We assume $\mathbb{E}[P] > 0$ and $\mathbb{E}[W] > 0$ without loss of generality since otherwise both online and offline costs are equal to zero and we have nothing to prove.

Let Z_n^{WSPR} be the random variable denoting the cost of WSPR on an instance of n jobs under the probabilistic conditions of the theorem. Let Z_n^* be the random variable denoting the optimal offline cost for $1|r_j, pmtn|\sum_j w_j C_j$. Finally, let $Z_n^{\mathcal{R}}$ be the random variable for the optimal cost of the relaxed problem $1|pmtn|\sum_j w_j C_j$, which is solved optimally by the WSPR heuristic; clearly $Z_n^{\mathcal{R}} \leq Z_n^*$.

At time R_n , the release date of the final job in the instance, assume that no processing has been done; clearly, this will only increase the online cost of WSPR. Therefore, under this assumption, at time R_n , the WSPR heuristic essentially sees the problem $1|pmtn|\sum_j w_j C_j$ (i.e., all release dates are equal to zero). Consequently, we have that

$$Z_n^{WSPR} \leq R_n \sum_{j=1}^n W_j + Z_n^{\mathcal{R}}. \quad (1)$$

Considering the ratio of online to offline costs, we have that

$$\begin{aligned} \frac{Z_n^{WSPR}}{Z_n^*} &\leq \frac{R_n \sum_{j=1}^n W_j + Z_n^{\mathcal{R}}}{Z_n^{\mathcal{R}}} \\ &= 1 + \frac{R_n \sum_{j=1}^n W_j}{Z_n^{\mathcal{R}}}. \end{aligned}$$

Let \mathcal{B}_j be the event that $P_j \geq \mathbb{E}[P] \wedge W_j \geq \mathbb{E}[W]$. We have that $\mathbb{P}[\mathcal{B}_j] = \alpha$ for some α . Let J denote the set of jobs having property \mathcal{B}_j . If we consider n jobs, $|J|$ is a binomial random variable with parameters n and α . By the Strong Law of Large Numbers, $|J|/n \rightarrow \alpha$, almost surely and, therefore, $J = \Theta(n)$, almost surely. Next, in order to compute a lower bound on $Z_n^{\mathcal{R}}$, we consider the processing of only the jobs in J . We re-order the indices on the W and P random variables in J such that

$$\frac{W_1}{P_1} \geq \frac{W_2}{P_2} \geq \dots \geq \frac{W_{|J|}}{P_{|J|}}.$$

Applying the WSPR heuristic to the jobs in J , we observe the following: The completion time of the first job processed, job 1, is P_1 ; the completion time of job 2 is $P_1 + P_2$; the completion time of job k is $P_1 + \dots + P_k$. Therefore, a lower bound for serving the set J of jobs, which is also a lower bound for $Z_n^{\mathcal{R}}$, is

$$\sum_{j=1}^{|J|} \mathbb{E}[W] \sum_{i=1}^j \mathbb{E}[P] = \mathbb{E}[W] \mathbb{E}[P] \frac{|J|(|J|+1)}{2}.$$

Using the fact that $|J| = \Theta(n)$, almost surely, we have that $Z_n^{\mathcal{R}} = \Omega(n^2)$, almost surely. By the Strong Law of Large Numbers, $\sum_{j=1}^n W_j = \Theta(n)$ almost surely and, by Lemma 2 with $r = 1$, $R_n = o(n)$ almost surely. We are therefore able to conclude that, as $n \rightarrow \infty$,

$$\frac{R_n \sum_{j=1}^n W_j}{Z_n^{\mathcal{R}}} \rightarrow 0,$$

almost surely and the proof is complete. ■

4.2 Online $1|r_j|\sum_j w_j C_j$

A non-preemptive version *Non-preemptive Weighted Shortest Processing Requirement* (NWSPR) of WSPR is easily defined: Whenever the machine is available to process a job, if there remain unprocessed jobs, choose the job with the highest ratio w_i/p_i . We are able to prove the exact same result as Theorem 4:

Theorem 5 *Under Assumptions 4, 5 and 6, if $\mathbb{E}[Y] < \infty$, then the NWSPR heuristic is almost surely asymptotically optimal for the online version of $1|r_j|\sum_j w_j C_j$.*

The proof of Theorem 5 is very similar to that of Theorem 4; we detail only the *differences*.

Proof Outline First, note that $Z_n^{\mathcal{R}}$ is the optimal value for both $1|pmtn|\sum_j w_j C_j$ and $1|\sum_j w_j C_j$, since these two problems are essentially identical. Equation (1) is modified to become

$$Z_n^{NWSPR} \leq \left(R_n + \max_{1 \leq i \leq n} P_i \right) \sum_{j=1}^n W_j + Z_n^{\mathcal{R}}.$$

The reason for this modification is because at time R_n , we can not relate NWSPR's actions to the problem $1|\sum_j w_j C_j$, since it might be busy processing some job. But after $\max_{1 \leq i \leq n} P_i$ time, we are certain that the machine has finished whatever job had been in progress at time R_n . Therefore, at time $R_n + \max_{1 \leq i \leq n} P_i$, assuming that no job has been processed, NWSPR “sees” the problem $1|\sum_j w_j C_j$. After recalling that $\max_{1 \leq i \leq n} P_i = o(n)$ almost surely (Lemma 2 with $r = 1$), the rest of the proof remains identical. ■

5 Parallel Machine Minsum Online Scheduling

We first consider online versions of the parallel machine scheduling problems $Q|r_j, pmtn|\sum_j C_j$ and $P|r_j|\sum_j C_j$; offline versions of both these problems are NP-hard. We show that well-known heuristics for these problems are asymptotically optimal, almost surely. We then study $Q|r_j, pmtn|\sum_j w_j C_j$ and $Q|r_j|\sum_j w_j C_j$ and show that, if we allow for non-polynomial time algorithms, there exist online algorithms that are asymptotically optimal, almost surely, for these difficult scheduling problems.

5.1 Online $Q|r_j, pmtn|\sum_j C_j$

Consider the *Shortest Remaining Processing Requirement on Fastest Machine* (SRPR-FM) heuristic, which is also an online algorithm: At any given time, the job with the shortest remaining processing requirement is assigned to the fastest machine, the job with the second shortest remaining processing requirement is assigned to the second fastest machine, and so on. Note that the SRPR-FM heuristic solves $Q|pmtn|\sum_j C_j$ exactly; e.g., see [27].

The reason that we only consider unit weights in this section is that even $P|pmtn|\sum_j w_j C_j$ is NP-hard and our technique for proving asymptotical optimality for a well-known heuristic would

break down. Our approach requires that SRPR-FM *exactly* solves the machine scheduling problem when all release dates are zero. Further details are given in the proof of Theorem 6.

Theorem 6 *Under Assumptions 4, 5, and m fixed, if $\mathbb{E}[Y] < \infty$, then the SRPR-FM heuristic is almost surely asymptotically optimal for the online version of $Q|r_j, pmtn|\sum_j C_j$.*

Proof We assume $\mathbb{E}[P] > 0$ without loss of generality since otherwise both online and offline costs are equal to zero and we have nothing to prove.

Let $Z_n^{SRPR-FM}$ be the random variable denoting the cost of SRPR-FM on an instance of n jobs under the probabilistic conditions of the theorem. Let Z_n^* be the random variable denoting the optimal offline cost for $Q|r_j, pmtn|\sum_j C_j$. Finally, let Z_n^R be the random variable for the optimal cost of the relaxed problem $Q|pmtn|\sum_j C_j$, which is solved optimally by the SRPR-FM heuristic; clearly $Z_n^R \leq Z_n^*$.

At time R_n , the release date of the final job in the instance, assume that no processing has been done; clearly, this will only increase the online cost of SRPR-FM. Therefore, under this assumption, at time R_n , the SRPR-FM heuristic essentially sees the problem $Q|pmtn|\sum_j C_j$. Consequently, we have that

$$Z_n^{SRPR-FM} \leq nR_n + Z_n^R. \quad (2)$$

The dependence of our proof on Equation (2) is the reason why we are limited to studying unit weights. Had we considered arbitrary weights, online algorithm SRPR-FM would encounter the NP-hard relaxation $Q|pmtn|\sum_j w_j C_j$ and we would not be able to construct a viable version of Equation (2).

Considering the ratio of online to offline costs, we have that

$$\begin{aligned} \frac{Z_n^{SRPR-FM}}{Z_n^*} &\leq \frac{nR_n + Z_n^R}{Z_n^R} \\ &= 1 + \frac{nR_n}{Z_n^R}. \end{aligned}$$

Next, we compute a lower bound on Z_n^R . Clearly, the optimal cost of $P|pmtn|\sum_j C_j$, where all machines have speed s_1 (the fastest speed), is a lower bound. Furthermore, the optimal cost of $1|pmtn|\sum_j C_j$, where the single machine has speed ms_1 , is a lower bound for $P|pmtn|\sum_j C_j$; the idea to consider a fast single machine relaxation was first considered by Eastman, Even, Isaacs [14]. Note that the non-preemptive *Shortest Processing Requirement*¹ (SPR) heuristic solves $1|pmtn|\sum_j C_j$ exactly (to see this, simply set $w_i = 1, \forall i$ in the introductory discussion of Section 4.1). We apply a similar argument to that in the proof of Theorem 4: Let \mathcal{B}_j be the event that $P_j \geq \mathbb{E}[P]$. We have that $\mathbb{P}[\mathcal{B}_j] = \alpha$ for some α . Let J denote the set of jobs having property \mathcal{B}_j . If we consider n jobs, $|J|$ is a binomial random variable with parameters n and α . By the Strong Law of Large Numbers, $|J|/n \rightarrow \alpha$, almost surely and, therefore, $J = \Theta(n)$, almost surely. Next, in order to compute a

¹Whenever the machine is available to process a job, choose the job with the shortest processing requirement.

lower bound on $Z_n^{\mathcal{R}}$, we consider the processing of only the jobs in J . We re-order the indices on the P random variables in J such that

$$P_1 \leq P_2 \leq \dots \leq P_{|J|}.$$

Applying the SPR heuristic to the jobs in J , we observe the following: The completion time of the first job processed, job 1, is $P_1/(ms_1)$; the completion time of job 2 is $(P_1 + P_2)/(ms_1)$; the completion time of job k is $(P_1 + \dots + P_k)/(ms_1)$. Therefore, a lower bound for serving the set J of jobs, which is also a lower bound for $Z_n^{\mathcal{R}}$, is

$$\sum_{j=1}^{|J|} \sum_{i=1}^j \frac{\mathbb{E}[P]}{ms_1} = \frac{\mathbb{E}[P]}{ms_1} \frac{|J|(|J| + 1)}{2}.$$

Using the fact that $|J| = \Theta(n)$, almost surely, we have that $Z_n^{\mathcal{R}} = \Omega(n^2)$, almost surely. Recalling that $R_n = o(n)$ almost surely, as $n \rightarrow \infty$,

$$\frac{nR_n}{Z_n^{\mathcal{R}}} \rightarrow 0,$$

almost surely and the proof is complete. ■

5.2 Online $P|r_j|\sum_j C_j$

Consider the non-preemptive *Shortest Processing Requirement* (SPR) heuristic, which is also an online algorithm: whenever a machine is available to process a job, choose the job with the shortest processing requirement. Note that the SPR heuristic solves $P|\sum_j C_j$ exactly; e.g., see [27]. Again, the reason that we only consider problems with unit weights is that $P|\sum_j w_j C_j$ is NP-hard. Our main result for this section is the following.

Theorem 7 *Under Assumptions 4, 5, and m fixed, if $\mathbb{E}[Y] < \infty$ then the SPR heuristic is almost surely asymptotically optimal for the online version of $P|r_j|\sum_j C_j$.*

The proof of Theorem 7 is very similar to that of Theorem 6; we detail only the *differences*.

Proof Outline Note that $Z_n^{\mathcal{R}}$ is the optimal value for $P|\sum_j C_j$. Equation (2) is modified to become

$$Z_n^{SPR} \leq n \left(R_n + \max_{1 \leq i \leq n} P_i \right) + Z_n^{\mathcal{R}}.$$

The reason for this modification is because at time R_n , we can not relate SPR's actions to the problem $P|\sum_j C_j$, since some machines might be busy processing some jobs. But after $\max_{1 \leq i \leq n} P_i$ time, we are certain that the machines have finished whatever jobs had been in progress at time R_n . Therefore, at time $R_n + \max_{1 \leq i \leq n} P_i$, assuming that no job has been processed, SPR "sees" the problem $P|\sum_j C_j$. The rest of the proof remains identical. ■

5.3 Online $Q|r_j, pmtn|\sum_j w_j C_j$ and $Q|r_j|\sum_j w_j C_j$

In this section, we point out that if we consider non-polynomial time online algorithms, we obtain asymptotic optimality results for more difficult machine scheduling problems. To illustrate our point, we consider $Q|r_j, pmtn|\sum_j w_j C_j$; similar reasoning applies to the non-preemptive version.

Let $A_{offline}$ be an offline algorithm that exactly solves $Q|pmtn|\sum_j w_j C_j$, which is NP-hard. Let A be the online algorithm that, whenever a new job is released, applies algorithm $A_{offline}$ to all known unprocessed jobs. The proofs of Theorems 4 and 6 can be combined to give the following result.

Theorem 8 *Under Assumptions 4, 5, 6 and m fixed, if $\mathbb{E}[Y] < \infty$, then online algorithm A is almost surely asymptotically optimal for the online version of $Q|r_j, pmtn|\sum_j w_j C_j$.*

Similarly, we also have the following result.

Theorem 9 *Under Assumptions 4, 5, 6 and m fixed, if $\mathbb{E}[Y] < \infty$, there exists an online algorithm \tilde{A} that is almost surely asymptotically optimal for the online version of $Q|r_j|\sum_j w_j C_j$.*

Acknowledgements

We thank the anonymous referees for their thoughtful comments, which improved the quality and clarity of our paper.

References

- [1] S. Albers and B. Schröder. An experimental study of online scheduling algorithms. *Journal of Experimental Algorithmics*, 7:3, 2002.
- [2] L. Allulli, G. Ausiello, and L. Laura. On the power of lookahead in on-line vehicle routing problems. In *Proceedings of the Eleventh International Computing and Combinatorics Conference*, pages 728–736, 2005.
- [3] E. Anderson and C. Potts. On-line scheduling of a single machine to minimize total weighted completion time. *Mathematics of Operations Research*, 29(3):686–697, 2004.
- [4] G. Bampis and E. Rouskas. On scheduling problems with applications to packet-switched optical WDM networks. Technical Report TR-2000-07, 2000. <http://citeseer.ist.psu.edu/bampis01scheduling.html>.
- [5] J. Beardwood, J. Halton, and J. Hammersley. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society*, 55:299–327, 1959.

- [6] A. Blum, P. Chalasani, D. Coppersmith, W. Pulleyblank, P. Raghavan, and M. Sudan. The minimum latency problem. In *Proceedings of the 26th ACM Symposium on the Theory of Computing*, 1994.
- [7] A. Bompadre, M. Dror, and J. Orlin. Probabilistic analysis of unit demand vehicle routing problems. *Journal of Applied Probability*, 44(1):259–278, 2007.
- [8] V. Bonifaci and L. Stougie. Online k-server routing problems. In *Proceedings of the 4th Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science*, 2006.
- [9] K. Chaudhuri, B. Godfrey, S. Rao, and K. Talwar. Paths, trees and minimizing latency. In *Proceeding of the 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003.
- [10] C. Chekuri, R. Motwani, B. Natarajan, and C. Stein. Approximation techniques for average completion time scheduling. *SIAM Journal on Computing*, 31:146–166, 2001.
- [11] G. Chen and Z. Shen. Probabilistic asymptotic analysis on stochastic online scheduling problems. *IIE Transactions*, 39(5):525–538, 2007.
- [12] C. Chou, M. Queyranne, and D. Simchi-Levi. The asymptotic performance ratio of an on-line algorithm for uniform parallel machine scheduling with release dates. *Mathematical Programming*, 106(1):137–157, 2006.
- [13] J. Correa and M. Wagner. LP-based online scheduling: from single to parallel machines. In *Proceedings of the 11th Integer Programming and Combinatorial Optimization Conference (IPCO)*, pages 196–209. Springer LNCS 3509, 2005.
- [14] W. Eastman, S. Even, and I. Isaacs. Bounds for the optimal scheduling of n jobs on m processors. *Management Science*, 11:268–279, 1964.
- [15] E. Feuerstein and L. Stougie. On-line single-server dial-a-ride problems. *Theoretical Computer Science*, 268(1):91–105, 2001.
- [16] J. Galambos. *The Asymptotic Theory of Extreme Order Statistics*. Robert E. Krieger Publishing Company, 1987.
- [17] M. Goemans, M. Queyranne, A. Schulz, M. Skutella, and Y. Wang. Single machine scheduling with release dates. *SIAM Journal on Discrete Mathematics*, 15:165–192, 2002.
- [18] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.
- [19] B. Hiller. Probabilistic competitive analysis of a dial-a-ride problem on trees under high load. Zib report 05-56, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2005.

- [20] P. Jaillet and M. Wagner. Online routing problems: value of advanced information as improved competitive ratios. *Transportation Science*, 40(2):200–210, 2006.
- [21] P. Jaillet and M. Wagner. Generalized online routing: New competitive ratios, resource augmentation and asymptotic analyses. *Operations Research*, 2007. Accepted for publication.
- [22] P. Kaminsky and D. Simchi-Levi. Asymptotic analysis of an on-line algorithm for the single machine completion time problem with release dates. *Operations Research Letters*, 29:141–148, 2001.
- [23] A.H.G. Rinnooy Kan. *Machine scheduling problems*. Martinus Nijhoff, The Hague, 1976.
- [24] S. Krumke, W. de Paepe, D. Poensgen, and L. Stougie. News from the online traveling repairman. *Theoretical Computer Science*, 295:279–294, 2003.
- [25] S. Krumke, W. de Paepe, D. Poensgen, and L. Stougie. Erratum to “news from the online traveling repairman”. *Theoretical Computer Science*, 352:347–348, 2006.
- [26] J.-C. Picard and M. Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26, 1978.
- [27] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, second edition, 2002.
- [28] M.W.P. Savelsbergh, R. Uma, and J. Wein. An experimental study of lp-based approximation algorithms for scheduling problems. *INFORMS Journal on Computing*, 17:123–136, 2005.
- [29] A. Schulz and M. Skutella. The power of α -points in preemptive single machine scheduling. *Journal of Scheduling*, 5:121–133, 2002.
- [30] A. Schulz and M. Skutella. Scheduling unrelated machines by randomized rounding. *SIAM Journal on Discrete Mathematics*, 15:450–469, 2002.
- [31] D. Simchi-Levi and O. Berman. Minimizing the total flow time of n jobs on a network. *IIE Trans.*, 23, 1991.
- [32] R. Sitters. *Complexity and approximation in routing and scheduling*. PhD thesis, Eindhoven University of Technology, 2004.
- [33] W. Smith. Various optimizers for single-stage production. *Naval Research and Logistics Quarterly*, 3:59–66, 1956.
- [34] J. Tsitsiklis. Special cases of traveling salesman and repairman problems with time windows. *Networks*, 22, 1992.
- [35] Y. Xia and D. Tse. Survey of single machine scheduling with application to web object transmission. Technical Report UCB/ERL M00/54, EECS Department, University of California, Berkeley, 2000. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2000/3908.html>.