

Primal-Dual Online Algorithms for the Parking Permit Problem

Christian Coester  

Department of Computer Science, University of Oxford, UK

Alex Turoczy  

Department of Computer Science, University of Oxford, UK

Abstract

The Parking Permit Problem (PPP), first studied by Meyerson, is a classic online problem generalizing the ski rental problem. We re-examine the PPP using the primal-dual scheme, obtaining simple algorithms with superior performance guarantees. Unlike previous work, which relied on reductions that degraded competitive ratios, we work with the problem's structure directly. We also provide near-matching lower bounds. Using the primal-dual framework, we find the PPP's deterministic competitive ratio exactly, and the randomized competitive ratio within an *additive* constant.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Online Algorithms, Competitive Analysis, Primal-Dual Algorithms, Parking Permit Problem

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.81

Funding Funded by the European Union (ERC, CCOO, 101165139). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

1 Introduction

The Parking Permit Problem (PPP) is a fundamental problem in online algorithms. It is the seminal *leasing* problem, meaning an algorithm may purchase a resource to satisfy constraints, but where the purchases last a fixed contiguous time duration, regardless of whether it is used in this time. In this paper, we reinvestigate this classic problem via the primal-dual framework, obtaining new and simpler algorithms with improved performance guarantees.

The PPP can be defined by the following canonical parking permit scenario. Suppose that you walk to work when the weather is good, and drive when it rains. When you drive, you must possess a valid parking permit, but there are K different types. Each permit type lasts a different duration and expires on a fixed day regardless of how many times it is utilized, where longer permits are cheaper on a per-day basis. The dilemma in the PPP is: what permit purchases should you make?

Essentially all prior work on PPP and variants [24, 11] involve a reduction to what we shall call the *Laminar* PPP. In the Laminar PPP, the permit cannot just be purchased starting from any arbitrary day. Instead, the permit durations form a laminar set family, meaning that two different permits' durations either do not overlap, or one is contained inside the other (see Figure 1). The reduction to the Laminar PPP loses a constant factor in the competitive ratio. In this paper, we provide techniques for solving leasing problems without assumptions of laminarity, leading to more direct algorithms with superior performance.



© Christian Coester and Alex Turoczy;

licensed under Creative Commons License CC-BY 4.0

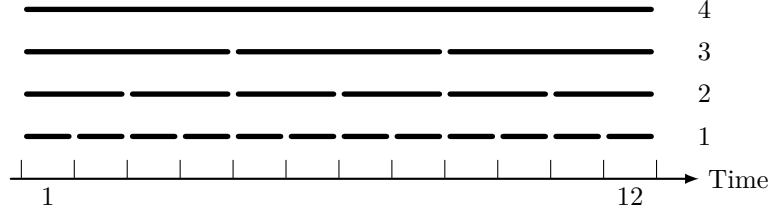
34th Annual European Symposium on Algorithms (ESA 2026).

Editors: Philip Bille, Seth Pettie, and Sabine Storandt; Article No. 81; pp. 81:1–81:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Illustration of the Laminar PPP with permit durations of length 1, 2, 4 and 12.

1.1 Our Results

Using primal-dual algorithms, we prove new upper and lower bounds for both deterministic and randomized algorithms. Our results in comparison to previous work are summarized in Table 1.

■ **Table 1** Summary of prior and new upper and lower bounds for the PPP.

		Deterministic PPP	Randomized PPP
Upper bounds	Previous	$4K$ [24]	$16 \ln K$ [24]
	New	K	$\ln K + \ln \ln K + O(1)$
Lower bounds	Previous	$K/6$ [24], 3 for $K = 3$ [26]	$\log_2(K)/2$ [24]
	New	K	$\ln K + \ln \ln K + o(1)$

In Section 2, we prove that the deterministic competitive ratio is exactly K , closing a factor 24 gap between prior upper and lower bounds. Our algorithm is simple and serves as a warm-up to the randomized algorithm.

► **Theorem 1.** *The deterministic competitive ratio for the online PPP is exactly K .*

Our randomized algorithm uses more sophisticated techniques. Buchbinder and Naor [6] previously observed that Meyerson’s $O(\log K)$ -competitive randomized algorithm can be recovered via the primal-dual framework, since the *Laminar* PPP is a covering problem with row-sparsity K . However, if we avoid the reduction to the laminar case and the corresponding constant factor loss, the central difficulty is that the linear program of the non-laminar version has unbounded row-sparsity, which means directly using generic online covering algorithms would result in unbounded competitive ratio for fixed K . Our main contribution is doing multiplicative updates over the *sum* of relevant variables, rather than the individual variables alone.

► **Theorem 2.** *The randomized competitive ratio for the online PPP is at most $\ln K + \ln \ln K + O(1)$ as $K \rightarrow \infty$.*

We also provide an improved lower bound, showing our algorithm’s competitive ratio only asymptotically differs from the lower bound by an *additive* constant of 2. In contrast, prior bounds left a *multiplicative* gap of $32 \ln 2 \approx 22.18$.

► **Theorem 3.** *Any randomized online algorithm for the PPP with K permits has a competitive ratio of at least $S_K = \ln K + \ln \ln K + o(1)$, defined by $S_1 := 1$ and $S_{K+1} := S_K + \frac{S_K}{\exp(S_K) - 1}$.*

We conjecture that this lower bound is exactly tight. At a high level, it is based on a recursive combination of ski rental lower bounds.

1.2 Related Work

1.2.1 The Parking Permit Problem

Several other online problems involving purchasing dilemmas preceded the PPP. The classic online ski rental problem is one of the most well-understood problems in online algorithms, encapsulating the dilemma of whether to rent or buy a particular resource. For this problem, the competitive ratio is 2 for deterministic algorithms, and $e/(e-1)$ for randomized algorithms [20]. It corresponds to the special case of the PPP where $K = 2$ and the more expensive permit has infinite duration.

The PPP was first introduced by Meyerson [24]. Whereas many other problems considered purchases of resources which then last forever, the PPP is one of the earliest problems using a *leasing optimization model*, where resources have a fixed lifespan. Meyerson presented upper and lower bounds for deterministic and randomized algorithms, which are matching up to multiplicative constants. In particular, the paper showed the competitive ratio for deterministic algorithms was between $\frac{K}{6}$ and $4K$, and the competitive ratio for randomized algorithms is between $\log_2(K)/2$ and $16 \ln K$. This work also introduced the laminar version of the problem, on which their algorithms and lower-bound results were based. Additionally, they showed applications to variants of Steiner forest problems.

A generalization of PPP, called the 2D PPP, was studied by Hu et al. [19]. In this problem, demands vary along two axes, the time and also over required capacities. This setting has applications in cloud computing, where for example the number of virtual machines required may vary over time based on the popularity of a website. This paper demonstrated that the competitive ratio of this problem for deterministic algorithms is $\Theta(K)$, which was achieved using pseudo-polynomial time algorithms. De Lima et al. [11] proposed another generalization of PPP, the Multi-PPP, in which the online demands can be any non-negative integer. They demonstrated a competitive ratio preserving pseudo-polynomial time reduction from the Laminar Multi-PPP to Laminar PPP, which involves running an arbitrary number of PPP instances concurrently. Therefore, Meyerson's previous results (which were also based on Laminar PPP) also hold for Multi-PPP. They also demonstrated that this could be turned into a strictly polynomial-time algorithm, while losing a factor of 2 in the competitiveness. They showed this technique can also convert Hu et al.'s 2D PPP algorithm into a strictly polynomial-time algorithm.

Since the introduction of the PPP, many other problems have been reformulated in the leasing optimization model, including Online Set Cover with Leasing [1] and Leasing Non-Metric Facility Location [23].

There has also been some work studying the PPP in learning-augmented settings. The first, by Kharchenko and Kononov [26], focused on the case of $K = 3$. A recent preprint by Ameli, Sanita and Venzin [3] proposed a general framework for learning-augmented algorithms for covering/packing problems, with the PPP as a special case, along with Set Cover, Non-Metric Facility Location, and Metric Facility Location. Coester, Tudose and Turoczy [9] present learning-augmented algorithms based on dual predictions for a number of online minimization problems, including the parking permit problem.

1.2.2 Primal-Dual Algorithms

For decades, LP duality theory has been utilized for the design and analysis of exact algorithms [16, 14, 15, 13, 10] and approximation algorithms [18, 22, 17, 2]. In the field of online algorithms, Buchbinder and Naor first developed a primal-dual framework for covering-packing problems across several papers [6, 7, 8]. In light of their work, it has

been observed that several other earlier papers can be re-interpreted from a primal-dual perspective [21, 25], leading to fruitful extensions [12, 8], including convex objectives [4], and new breakthroughs on problems such as weighted paging [5].

1.3 Preliminaries

We write $\text{alg}(I)$ for the cost of an algorithm alg on instance I , and similarly $\text{opt}(I)$ for the optimal offline algorithm. The algorithm is ρ -competitive if $\text{alg}(I) \leq \rho \cdot \text{opt}(I)$ for any instance I , and the minimal such ρ is also called the competitive ratio. We will often drop I from the notation when it is clear from the context and write alg and opt for both the names of the algorithms and their cost.

For $k = 1, \dots, K$, we denote by $D_k \in \mathbb{Z}_{>0}$ the duration and by $C_k \in \mathbb{R}_{>0}$ the cost of permit type k . A sequence of days $\sigma = (\sigma_1, \dots, \sigma_T)$ is revealed online, where $\sigma_t = 1$ denotes that day t is rainy and $\sigma_t = 0$ otherwise.

By letting $x_{k,t}$ indicate the algorithm's decision to buy a permit of type k on day t , we introduce a formulation of the problem as an integer linear program as follows.

$$\begin{aligned} \text{Minimize} \quad & \sum_{t=1}^T \sum_{k=1}^K C_k x_{k,t} \\ \text{subject to} \quad & \sum_{k=1}^K \sum_{s=t-D_k+1}^t x_{k,s} \geq \sigma_t, \quad t = 1, \dots, T \\ & x_{k,t} \in \mathbb{Z}_{\geq 0}, \quad t = 1, \dots, T, k = 1, \dots, K \end{aligned} \tag{1}$$

By relaxing the integrality constraint of (1), we arrive at a primal covering linear program and its corresponding dual packing program. We will call the formulation of (2) the *Fractional PPP*.

Primal:

$$\begin{aligned} \text{Minimize} \quad & \sum_{t=1}^T \sum_{k=1}^K C_k x_{k,t} \\ \text{subject to} \quad & \sum_{k=1}^K \sum_{s=t-D_k+1}^t x_{k,s} \geq \sigma_t, \quad t = 1, \dots, T \\ & x_{k,t} \geq 0, \quad t = 1, \dots, T, k = 1, \dots, K \end{aligned}$$

(2)

Dual:

$$\begin{aligned} \text{Maximize} \quad & \sum_{t=1}^T \sigma_t y_t \\ \text{subject to} \quad & \sum_{s=t}^{t+D_k-1} y_s \leq C_k, \quad t = 1, \dots, T, k = 1, \dots, K \\ & y_t \geq 0, \quad t = 1, \dots, T \end{aligned}$$

(3)

One way to interpret the dual program is the following. Each variable y_t can be interpreted as representing the amount of blame we assign to day t for the cost of the PPP instance. The dual constraint $\sum_{s=t}^{t+D_k-1} y_s \leq C_k$ is then interpreted as ensuring that the assignment of blame is consistent; we cannot blame more than C_k to a contiguous sequence of D_k days, as we could cover this sequence with a single permit costing C_k .

2 Warm-Up: Deterministic Algorithms

We begin by presenting our deterministic online algorithm and lower bound, which serve as a warm-up to the randomized results in the subsequent section.

2.1 Deterministic Primal-Dual Algorithm

We provide a simple K -competitive primal-dual algorithm for the deterministic PPP, given in Algorithm 1.

Algorithm 1 K -competitive deterministic PPP algorithm.

-
- 1: Initialize $\mathbf{x} \leftarrow \mathbf{0}$ and $\mathbf{y} \leftarrow \mathbf{0}$.
 - 2: **for** each rainy day t not covered by any permit **do**
 - 3: Increase y_t continuously until $\sum_{s=t-D_k+1}^t y_s = C_k$ for some k .
 - 4: Purchase such permit k , i.e., set $x_{k,t} = 1$.
 - 5: **end for**
-

► **Lemma 4.** *Algorithm 1 is a K -competitive algorithm for the PPP.*

Proof. Let P and D represent the primal and dual values, respectively. We prove the following claims.

1. Algorithm 1 produces a feasible online primal solution $\mathbf{x}$.
2. At all points in the algorithm $P \leq K \cdot D$
3. Algorithm 1 produces a feasible dual solution $\mathbf{y}$.

Claim 1 is clear, as all rainy days will eventually be covered by a permit, because by increasing y_t , eventually $\sum_{s=t-D_k+1}^t y_s$ will reach some C_k .

Claim 2 holds because the maximum amount spent on a fixed permit type k is at most D . To see this, observe that we only purchase type k at time t if $\sum_{s=t-D_k+1}^t y_s = C_k$, but then we do not buy any new permits for D_k days. Therefore, each purchase of cost C_k for a type k permit can be uniquely associated with a dual cost of C_k . Any time we make a purchase of a type k permit costing C_k , we can charge this to the dual cost of C_k , and each charge must be unique as each type k permit purchase is separated from the next by a time duration of D_k . This proves the claim.

Claim 3 is evident, as no dual constraint can be violated before buying a permit.

With these claims proven, we appeal to weak duality to obtain that

$$\text{alg} = \sum_{s=1}^T \sum_{k=1}^K x_{k,t} \leq K \cdot \sum_{s=1}^T y_s \leq K \cdot \text{opt}. \quad \blacktriangleleft$$

2.2 Deterministic Lower Bound

In this section, we prove a lower bound on the deterministic competitive ratio of the PPP, which matches our upper bound of K .

► **Lemma 5.** *The deterministic competitive ratio for the PPP is at least K*

Proof. We choose N to be some large integer constant. For $1 \leq k \leq K$, we take $C_k = N^{k-1}$ and $D_k = N^{2(k-1)}$. The adversarial sequence is such that a day is rainy exactly when alg has no valid permit at the start of said day. The input of rainy days terminates once a type K permit has been purchased or once D_K days have elapsed, whichever occurs first. We assume without loss of generality that alg only possesses at most one permit at a time. Let us define the problem instance with K permits by I_K .

We will show by induction on K that for any online algorithm alg, we must have $\text{alg}(I_K) \geq (K - \epsilon_{K,N})\text{opt}(I_K)$, where $\epsilon_{K,N} \rightarrow 0$ as $N \rightarrow \infty$. For $K = 1$, this is trivial because the online algorithm cannot perform better than the offline optimal algorithm. We move on to the inductive step, showing that if the result holds up to K , then it holds for $K + 1$ too. We proceed via a case analysis.

Case 1: alg does not buy a type $K + 1$ permit during I_{K+1} . In this case, alg will only use permits of type K or less. Observe that due to the adversarial input, the algorithm must have a permit for every day during I_{K+1} . The cheapest option per day is the type K permit, as $\frac{C_k}{D_k} = \frac{N^{k-1}}{N^{2(k-1)}} = \frac{1}{N^{k-1}}$ is strictly decreasing with k . As there are $D_{K+1} = N^{2K}$ days, this means that $\text{alg} \geq \frac{1}{N^{K-1}} \cdot N^{2K} = N^{K+1}$. Observe however that $\text{opt} \leq C_{K+1} = N^K$, as one strategy for this interval is just to buy the permit of type $K + 1$. Hence we have $\text{alg} \geq N^{K+1} \geq N \cdot \text{opt}$. For N large enough, this satisfies the required bound.

Case 2: alg does buy a type $K + 1$ permit during I_{K+1} . Note that I_{K+1} is just a concatenation of several (complete) I_K instances and a single (possibly incomplete) instance at the end during which alg buys the type $K + 1$ permit. On the prior complete instances, the algorithm's cost is at most $\text{alg} - C_{K+1}$ and the optimal cost is at least $\text{opt} - C_K$, because the final incomplete instance contributes at most C_K to opt's cost as it could cover this instance with a single permit of type C_K . Thus, applying the induction hypothesis to all complete instances yields

$$\text{alg} - C_{K+1} \geq (K - \epsilon_{K,N}) \cdot (\text{opt} - C_K).$$

Using $\text{opt} \leq C_{K+1}$ (because one solution to I_{K+1} is to simply buy the type $K + 1$ permit immediately) and rearranging, we get

$$\text{alg} + KC_K \geq (K - \epsilon_{K,N} + 1) \cdot \text{opt}.$$

Because $\text{alg} \geq C_{K+1} = N \cdot C_K$, the left hand side is at most $\text{alg} \cdot (1 + \frac{K}{N})$. Thus,

$$\frac{\text{alg}}{\text{opt}} \geq \frac{K - \epsilon_{K,N} + 1}{1 + \frac{K}{N}} \rightarrow K + 1 \quad \text{as } N \rightarrow \infty,$$

which concludes the induction step. ◀

3 Randomized Algorithms

In this section, we develop a randomized online algorithm using the primal-dual framework. We also show an improved lower bound on the randomized competitive algorithm, which is within an *additive* constant of our algorithm's competitive ratio.

3.1 Randomized Primal-Dual Algorithm

In this section, we develop a near-optimal randomized online algorithm, satisfying the guarantee of the following lemma:

► **Lemma 6.** *The randomized competitive ratio for the PPP is at most $\ln K + \ln \ln K + O(1)$.*

In Meyerson's [24] randomized online algorithm, the most significant deterioration to the competitive ratio occurs due to the assumption of laminarity. In this section, we develop novel algorithmic techniques which show how the non-laminar structure can be handled directly, without factor-losing reductions to the laminar case.

Firstly, we present Algorithm 2, a primal-dual online algorithm to the *fractional* PPP, which we show in Lemma 7 achieves a desirable competitive ratio. Afterwards, we show that this can be rounded into an integral solution online, without degrading the competitive ratio.

Algorithm 2 draws inspiration from previous primal-dual algorithms [8], such as those for set cover. In many primal-dual algorithms, each primal variable increases as an exponential function of some corresponding dual variables. For problems like set cover, this results in a competitive ratio logarithmic in the maximum number of sets containing any element (corresponding to the row sparsity in general covering problems). However, when considering non-laminar PPP as an instance of set cover, even for fixed K this number is unbounded. Instead, on each day t , we orchestrate the increase to variables $x_{k,t}$ instead so that the summation $\sum_{s=t-D_k+1}^t x_{k,s}$ increases exponentially with the dual variables. As shall be shown, this results in a competitive ratio being a function of K .

■ **Algorithm 2** Online Fractional Algorithm for the fractional PPP.

```

1: Initialize all  $x_{k,t}$  and  $y_t$  to 0, for  $1 \leq k \leq K, 1 \leq t \leq T$ .
2: for each rainy day  $t$  do
3:   while  $\sum_{k=1}^K \sum_{s=t-D_k+1}^t x_{k,s} < 1$  do
4:     Increase  $y_t$  uniformly.
5:     For all  $k$ ,  $x_{k,t} := \left( \frac{1}{K \ln K} + \sum_{s=t-D_k+1}^{t-1} x_{k,s} \right) \left( \exp \left( \frac{y_t}{C_k} \right) - 1 \right)$ 
6:   end while
7: end for

```

► **Lemma 7.** *Algorithm 2 is $(\ln K + \ln \ln K + 2 + o(1))$ -competitive for fractional PPP.*

Proof. Let P and D represent the primal and dual objective values throughout the algorithm. We prove this result in 3 steps.

1. Algorithm 2 produces a feasible online primal solution $\mathbf{x}$.
2. At all times in the algorithm $P \leq (1 + \frac{1}{\ln K})D$
3. Algorithm 2 produces a dual solution $\mathbf{y}$ which violates each constraint by a factor no greater than $\ln(1 + K \ln K)$.

For Claim 1, observe that throughout the while loop, $x_{k,t}$ strictly increases, so eventually $\sum_{k=1}^K \sum_{s=t-D_k+1}^t x_{k,s} = 1$ is satisfied. Additionally, non-negativity is satisfied because at $y_t = 0$, $x_{k,t} = 0$, so $x_{k,t} = 0$ increases continuously and is always non-negative.

Moving to Claim 2, observe that this is clearly satisfied initially, as $P = D = 0$. We are left with showing that for a rainy day t , $\frac{\partial P}{\partial y_t} \leq (1 + \frac{1}{\ln K}) \frac{\partial D}{\partial y_t}$. Observe

$$\begin{aligned}
\frac{\partial P}{\partial y_t} &= \sum_{k=1}^K C_k \frac{\partial x_{k,t}}{\partial y_t} \\
&= \sum_{k=1}^K C_k \left(\frac{1}{K \ln K} + \sum_{s=t-D_k+1}^{t-1} x_{k,s} \right) \cdot \frac{1}{C_k} \cdot \exp\left(\frac{y_t}{C_k}\right) \\
&= \sum_{k=1}^K \left(\frac{1}{K \ln K} + \sum_{s=t-D_k+1}^{t-1} x_{k,s} \right) \cdot \left(\exp\left(\frac{y_t}{C_k}\right) - 1 + 1 \right) \\
&= \sum_{k=1}^K \left(x_{k,t} + \frac{1}{K \ln K} + \sum_{s=t-D_k+1}^{t-1} x_{k,s} \right) \\
&= \sum_{k=1}^K \left(\frac{1}{K \ln K} + \sum_{s=t-D_k+1}^t x_{k,s} \right) \\
&\leq \frac{1}{\ln K} + 1 \\
&= \left(1 + \frac{1}{\ln K} \right) \cdot \frac{\partial D}{\partial y_t}
\end{aligned}$$

We finally turn to Claim 3. We will show that $\mathbf{y}$ cannot violate any (k, t) -constraint by more than a factor of $\ln(1 + K \ln K)$. Fix (k, t) . We seek to show that $\sum_{s=t-D_k+1}^t y_s \leq C_k \cdot \ln(1 + K \ln K)$. We claim for all t' with $t - D_k + 1 \leq t' \leq t$ that

$$\sum_{s=t-D_k+1}^{t'} x_{k,s} \geq \frac{1}{K \ln K} \left(\exp\left(\frac{\sum_{s=t-D_k+1}^{t'} y_s}{C_k}\right) - 1 \right) \quad (4)$$

We prove this by induction on t' . The base case of $t' = t - D_k + 1$ is immediate from the definition of $x_{k,t-D_k+1}$ and nonnegativity of $\mathbf{x}$.

For $t - D_k + 1 < t' \leq t$, we use the following inductive argument:

$$\begin{aligned}
\sum_{s=t-D_k+1}^{t'} x_{k,s} &= x_{k,t'} + \sum_{s=t-D_k+1}^{t'-1} x_{k,s} \\
&\geq \left(\frac{1}{K \ln K} + \sum_{s=t-D_k+1}^{t'-1} x_{k,s} \right) \left(\exp\left(\frac{y_{t'}}{C_k}\right) - 1 \right) + \sum_{s=t-D_k+1}^{t'-1} x_{k,s} \\
&= \frac{1}{K \ln K} \left(\exp\left(\frac{y_{t'}}{C_k}\right) - 1 \right) + \sum_{s=t-D_k+1}^{t'-1} x_{k,s} \exp\left(\frac{y_{t'}}{C_k}\right) \\
&\geq \frac{1}{K \ln K} \left(\exp\left(\frac{y_{t'}}{C_k}\right) - 1 \right) \\
&\quad + \frac{1}{K \ln K} \left(\exp\left(\frac{\sum_{s=t-D_k+1}^{t'-1} y_s}{C_k}\right) - 1 \right) \exp\left(\frac{y_{t'}}{C_k}\right) \\
&= \frac{1}{K \ln K} \left(\exp\left(\frac{\sum_{s=t-D_k+1}^{t'} y_s}{C_k}\right) - 1 \right)
\end{aligned}$$

as required. Specializing (4) for $t' = t$, and observing by the algorithm that $\sum_{s=t-D_k+1}^t x_{k,s} \leq 1$, we find that $\frac{1}{K \ln K} \left(\exp\left(\frac{\sum_{s=t-D_k+1}^t y_s}{C_k}\right) - 1 \right) \leq 1$, which can be rearranged into $\sum_{s=t-D_k+1}^t y_t \leq C_k \cdot \ln(1 + K \ln K)$, as desired.

■ **Algorithm 3** Deterministic fractional PPP to randomized integral PPP conversion.

Require: Fractional online algorithm alg_{frac} satisfying assumptions (1) and (2).

- 1: Initiate online fractional solution $\mathbf{x} \leftarrow \mathbf{0}$
- 2: **for** each rainy day t not covered by any permit **do**
- 3: Use alg_{frac} to compute fractional online solution $\mathbf{x}$ up to day t .
- 4: Purchase exactly one of the permits at random, where permit k is chosen with probability $\frac{x_{k,t}}{\sum_{k'} x_{k',t}}$.
- 5: **end for**

From the above claims, the competitive bound follows quickly. From Claim 3, we know that $\hat{\mathbf{y}} := \frac{\mathbf{y}}{\ln(1+K \ln K)}$ is a feasible dual solution. Weak duality implies that the objective value of $\hat{\mathbf{y}}$ is a lower bound on the optimal primal solution value. Hence, $\frac{1}{\ln(1+K \ln K)} \sum_{t=1}^T y_t \leq \text{opt}$. By Claim 2, we hence know that

$$\text{alg} \leq \left(1 + \frac{1}{\ln K}\right) \sum_{t=1}^T y_t \leq \left(1 + \frac{1}{\ln K}\right) (1 + \ln(1 + K \ln K)) \text{opt}$$

Then, observing that $\ln(1 + K \ln K) = \ln K + \ln \ln K + o(1)$ we have

$$\begin{aligned} \left(1 + \frac{1}{\ln K}\right) (1 + \ln(1 + K \ln K)) &= \left(1 + \frac{1}{\ln K}\right) (1 + \ln K + \ln \ln K + o(1)) \\ &= 2 + \ln K + \ln \ln K + o(1), \end{aligned}$$

concluding the proof. ◀

Next, we show that there exists an online rounding algorithm for the fractional PPP, which creates a randomized integral solution whose expected cost is no greater than the fractional solution. Paired with Lemma 7, this immediately proves Lemma 6. Meyerson [24] proved a similar result, however the reduction was not perfect, as the conversion from a deterministic fractional PPP to randomized PPP lost a factor of 8 in the competitive ratio.

► **Lemma 8.** *There exists an online randomized rounding algorithm, which takes as input an online instance σ and an online algorithm for the fractional PPP with cost alg_{frac} , and constructs a randomized online integral solution to the PPP with cost alg such that $\mathbb{E}[\text{alg}] \leq \text{alg}_{\text{frac}}$.*

Proof. Fix an online instance σ , and suppose we have an online algorithm alg_{frac} for the fractional PPP. We assume that (1) on day t , alg_{frac} may only increase the variables $x_{k,t}$, and that (2) rainy days are covered by a total mass of 1 permit. These assumptions are without loss of generality, as any alg_{frac} can be converted into an algorithm satisfying this without increasing the objective value. We claim that Algorithm 3 is then a randomized integral solution with $\mathbb{E}[\text{alg}] \leq \text{alg}_{\text{frac}}$.

Firstly, observe that the algorithm always produces a feasible integral solution, as every day which begins without a permit will have a new permit purchased. Next, let $Z_{k,t}$ be a random indicator variable which indicates whether the permit of type k is purchased at time t by the algorithm. Then, we have that $\text{alg} = \sum_k C_k \sum_{t=1}^T Z_{k,t}$. To prove that $\mathbb{E}[\text{alg}] \leq \text{alg}_{\text{frac}}$, it therefore suffices to show that $\mathbb{E}[Z_{k,t}] \leq x_{k,t}$ for all k and t by linearity of expectation. The inequality is obvious for days t with $\sigma_t = 0$, because in that case both sides of the equation equal 0. Focusing instead on days with $\sigma_t = 1$, let t_j denote the j th day with $\sigma_t = 1$. We prove by induction on j that $\mathbb{E}[Z_{k,t_j}] = x_{k,t_j}$.

For the base case, consider $j = 1$. Observe that $\sum_k x_{k,t_1} = 1$ as $x_{k,t} = 0$ for times $t < t_1$, and day t_1 must be covered on day t_1 for the fractional solution to be feasible. Therefore, on day t_1 , the algorithm purchases permit k with probability $\frac{x_{k,t_1}}{\sum_{k'} x_{k',t_1}} = x_{k,t_1}$. This means $\mathbb{E}[Z_{k,t_1}] = x_{k,t_1}$, thereby demonstrating the base case.

For the inductive step, fix k and j , and assume the claim holds for all $j' < j$ and for all k . In order for t_j to be a rainy day where Z_{k,t_j} even has the possibility of being non-zero, it must be that alg purchased a permit which became invalid in $(t_{j-1}, t_j]$. Let I_j be the event that alg owned a permit which became invalid at time $(t_{j-1}, t_j]$. The necessity of I_j for $Z_{k,t_j} = 1$ implies

$$\mathbb{E}[Z_{k,t_j}] = \mathbb{P}[Z_{k,t_j} = 1] = \mathbb{P}[Z_{k,t_j} = 1 \cap I_j] = \mathbb{P}[Z_{k,t_j} = 1 | I_j] \cdot \mathbb{P}[I_j]. \quad (5)$$

Now, $\mathbb{P}[Z_{k,t_j} = 1 | I_j] = \frac{x_{k,t_j}}{\sum_{k'} x_{k',t_j}}$ by definition of Algorithm 3. On the other hand, by additivity of disjoint events and the inductive hypothesis, we get

$$\mathbb{P}[I_j] = \sum_{k=1}^K \sum_{t_{j-1} < s \leq t_j} \mathbb{E}[Z_{k,s-D_k}] = \sum_{k=1}^K \sum_{t_{j-1} < s \leq t_j} x_{k,s-D_k}. \quad (6)$$

However, we have that $\sum_{k=1}^K \sum_{t_{j-1} < s \leq t_j} x_{k,s-D_k} = \sum_{k'} x_{k',t_j}$, because the mass of permits which become invalid in $(t_{j-1}, t_j]$ must be made up by new mass to permits at time t_j . Therefore, we have $\mathbb{E}[Z_{k,t_j}] = \frac{x_{k,t_j}}{\sum_{k'} x_{k',t_j}} \cdot \sum_{k'} x_{k',t_j} = x_{k,t_j}$, which finishes the inductive step and hence the proof. $\blacktriangleleft$

3.2 Randomized Lower bound

We now provide a lower bound to the fractional PPP, matching the upper bound guarantee of Algorithm 2 up to an additive constant. Since any randomized algorithm can be turned into a fractional algorithm with the same competitive ratio, by defining $x_{k,t}$ as the probability of purchasing permit k at time t , this implies the lower bound of Theorem 3 for randomized algorithms,

We begin by showing that it suffices to lower bound the laminar version of the problem. Recall that in Laminar PPP, we require that D_k/D_{k-1} is an integer for all $k = 2, \dots, K$, and any permit of type k expires on the next day that is a multiple of D_k . The linear program for the laminar version is the same as (2) except that variables $x_{k,t}$ exist only for $t = 1 + (i-1)D_k$ for some integer i (i.e., all other $x_{k,t}$ are set to 0).

► **Lemma 9.** *For fixed K , the competitive ratio for online fractional PPP is at least the competitive ratio for online fractional Laminar PPP.*

Proof. Fix K . Suppose there is an α -competitive algorithm alg' for the online fractional (non-laminar) PPP. We will show that there is an α -competitive algorithm alg for Laminar PPP.

Consider an instance for Laminar PPP with durations $D_1, \dots, D_K$, costs $C_1, \dots, C_K$ and an online sequence σ . We may assume without loss of generality that $D_1 = 1$. This implies that each day t can be uniquely specified by a tuple $(i_1, \dots, i_K)$ with $1 \leq i_k \leq D_{k+1}/D_k$ for $k < K$, such that day t is covered by the i_K -th type K permit, and covered by the i_{K-1} -th permit among the D_K/D_{K-1} permits of type $K-1$ overlapping this type K permit, and so on. Formally, this mapping from tuple $(i_1, \dots, i_K)$ to day t is given by

$$f(i_1, \dots, i_K) = 1 + \sum_{k=1}^K (i_k - 1)D_k.$$

Algorithm alg will work by simulating alg' on a modified non-laminar instance with durations D'_k , costs C'_k and an online sequence σ' . We will define $D'_k = 2^{k-1}D_k$, so that an interval of type k can keep twice as many intervals of type $k-1$ as it originally could. The costs C'_k are kept identical to C_k . An online sequence σ for the original laminar instance is modified into an online sequence σ' as follows: if $t = f(i_1, \dots, i_K)$ is a rainy day in σ , then the day t' defined by $t' = f'(2i_1, \dots, 2i_K)$ is rainy in σ' , where f' is defined like f but with D_k replaced by D'_k . Let opt and opt' be the optimal solution values for σ and σ' , respectively. One may immediately observe that $\text{opt}' \leq \text{opt}$, as we can convert a laminar solution for σ into a laminar solution for σ' without increasing the cost, and the optimal non-laminar solution for σ' is at least this good.

We now wish to show $\text{alg} \leq \text{alg}'$, where alg and alg' denote the costs of the online algorithms' solutions for σ and σ' , respectively. Algorithm alg works by simulating alg' on σ' , where on a rainy day t , alg increases the weight of the permit of each type k covering t so that the mass of the type k permit covering t equals the total mass of type k permits covering t' by alg' . Observe that alg is clearly feasible.

We charge the cost of purchases by alg to the corresponding purchases by alg' that triggered them. To show that the cost of alg is at most that of alg' , it suffices to show that if two days $t_1 \neq t_2$ are covered by alg using separate permits of type k , then they charge to distinct purchases by alg' . Indeed, if $t_1 = f(i_1, \dots, i_K)$ and $t_2 = f(j_1, \dots, j_K)$, then the corresponding days $t'_1 = f'(2i_1, \dots, 2i_K)$ and $t'_2 = f'(2j_1, \dots, 2j_K)$ are a distance of at least D'_k apart. Thus, alg' cannot cover them with a single permit of type k .

Hence, we have $\frac{\text{alg}}{\text{opt}} \leq \frac{\text{ALG}'}{\text{OPT}'}$, which proves the lemma. $\blacktriangleleft$

In Lemma 11, we show that the randomized competitive ratio is lower bounded by S_K defined by the recurrence $S_1 := 1$ and $S_{K+1} := S_K + \frac{S_K}{\exp(S_K) - 1}$. We remark that $S_2 = \frac{e}{e-1}$ is the known optimal randomized competitive ratio for ski rental [20]. In fact, our lower bound is obtained by combining ski rental lower bounds recursively. We prove in Lemma 10 that S_K asymptotically matches the upper bound of Lemma 7 up to additive constants. We leave its proof to Appendix A.

► **Lemma 10.** *Let S_K be defined by $S_1 := 1$ and $S_{K+1} := S_K + \frac{S_K}{\exp(S_K) - 1}$. Then $S_K = \ln K + \ln \ln K + o(1)$ as $K \rightarrow \infty$.*

► **Lemma 11.** *Any online algorithm for the fractional online PPP with K permits has a competitive ratio of at least S_K , where $S_1 := 1$ and $S_{K+1} := S_K + \frac{S_K}{\exp(S_K) - 1}$.*

Proof. By Lemma 9 we may prove the lemma for the Laminar PPP. We proceed by induction on K . The base case $K = 1$ is obvious. Now considering $K \geq 2$, assume that for $K-1$ it is known that we have an adversarial strategy so that the competitive ratio is at least S_{K-1} . Let $C_k = N^k$ and $D_k = M^k$, where M and N are large integers and $M > N$. In particular, any type K interval consists of M disjoint type $K-1$ intervals. We construct an instance of length $T = D_K$, so that the entire problem instance consists of only one type K interval.

We define our adversarial instance for K as follows. We iteratively apply the S_{K-1} adversarial case (assumed to exist by the inductive hypothesis) to the i th type $K-1$ interval, for $i = 1, 2, \dots$. Let alg_i and opt_i denote cost of the online algorithm alg on the i th type $K-1$ interval, and opt_i the optimal offline cost on that interval.

If after the m th type $K-1$ interval we have $\sum_{i=1}^m \text{alg}_i \geq S_K \min\{\sum_{i=1}^m \text{opt}_i, C_K\}$, then we stop the problem instance, and observe that the algorithm on this instance would have a competitive ratio of S_K , as $\text{opt} \leq \min\{\sum_{i=1}^m \text{opt}_i, C_K\}$ in that case. As this would finish the proof, instead suppose that we have

$$\sum_{i=1}^m \text{alg}_i \leq S_K \min \left\{ \sum_{i=1}^m \text{opt}_i, C_K \right\} \quad (7)$$

at all times. Let $t_i = i \cdot D_{K-1}$ denote the final time step in the i th type $K-1$ permit. Also let $x_K(t_i)$ denote the mass of permit K which has been purchased at time t_i . Now, observe that the cost of the algorithm at time t_m is bounded by

$$C_K x_K(t_m) + \sum_{i=1}^m S_{K-1} \cdot \text{opt}_i (1 - x_K(t_i)) \leq \sum_{i=1}^m \text{alg}_i \leq S_K \min \left\{ \sum_{i=1}^m \text{opt}_i, C_K \right\} \quad (8)$$

We will use inequality (8) to bound each $x_K(t_i)$ from above and thereby arrive at a lower bound on the cost of alg. By making $M \gg N \gg 1$ arbitrarily large, meaning each type $K-1$ interval is a vanishingly narrow part of the type K permit, this PPP adversarial example becomes arbitrarily close to a continuous problem. Moreover, the minimum on the right-hand side will be C_K , and $x_K(t_m)$ will tend to 1 as otherwise the left-hand side would grow unbounded. By letting y be the continuous analog of $\sum_{i=1}^m \text{opt}_i$, the minimum possible cost of alg is thus lower bounded by the solution to the following optimization problem:

$$\min_{x_K} \quad C_K + S_{K-1} \int_0^{C_K} (1 - x_K(y)) dy \quad (9)$$

subject to

$$C_K x_K(y) + S_{K-1} \int_0^y (1 - x_K(s)) ds \leq S_K \cdot y \quad \forall y \in [0, C_K] \quad (10)$$

and $x_K(0) = 0$, x_K non-decreasing, $x_K(y) \in [0, 1]$. The minimum in (9) is achieved by maximizing the $x_K(y)$, and subject to the constraint (10), $x_K(y)$ is maximized by greedily maximizing $x_K(s)$ for all $s \in [0, y]$. Thus, the constraint (10) is tight in the optimal solution, we can solve for $x_K(y)$ by differentiating both sides of the constraint with respect to y :

$$C_K x'_K(y) + S_{K-1} (1 - x_K(y)) = S_K$$

Rearranging and using the recurrence definition of S_K :

$$C_K x'_K(y) = S_K - S_{K-1} + S_{K-1} x_K(y) = \frac{S_{K-1}}{\exp(S_{K-1}) - 1} + S_{K-1} x_K(y)$$

Given the initial condition $x_K(0) = 0$, this differential equation is solved by

$$x_K(y) = \frac{1}{\exp(S_{K-1}) - 1} \left(\exp\left(\frac{S_{K-1}y}{C_K}\right) - 1 \right).$$

Combining the lower bound (9) on the algorithm's cost with the fact that $\text{opt} \leq C_K$, the competitive ratio satisfies:

$$\frac{\text{alg}}{\text{opt}} \geq 1 + \frac{S_{K-1}}{C_K} \int_0^{C_K} (1 - x_K(y)) dy$$

We now evaluate the integral:

$$\begin{aligned}
 \int_0^{C_K} (1 - x_K(y)) dy &= \int_0^{C_K} \left(1 - \frac{1}{\exp(S_{K-1}) - 1} \left(\exp\left(\frac{S_{K-1}y}{C_K}\right) - 1 \right) \right) dy \\
 &= C_K + \frac{1}{\exp(S_{K-1}) - 1} \left(C_K - \frac{C_K}{S_{K-1}} (\exp(S_{K-1}) - 1) \right) \\
 &= C_K + \frac{C_K}{\exp(S_{K-1}) - 1} - \frac{C_K}{S_{K-1}} \\
 &= C_K \left(1 + \frac{1}{\exp(S_{K-1}) - 1} - \frac{1}{S_{K-1}} \right)
 \end{aligned}$$

Substituting back:

$$\begin{aligned}
 \frac{\text{alg}}{\text{opt}} &\geq 1 + S_{K-1} \left(1 + \frac{1}{\exp(S_{K-1}) - 1} - \frac{1}{S_{K-1}} \right) \\
 &= S_{K-1} + \frac{S_{K-1}}{\exp(S_{K-1}) - 1} \\
 &= S_K
 \end{aligned}$$

where the last equality is the recurrence defining S_K . This completes the induction. $\blacktriangleleft$

References

- 1 Sebastian Abshoff, Christine Markarian, and Friedhelm Meyer auf der Heide. Randomized online algorithms for set cover leasing problems. In *Combinatorial Optimization and Applications: 8th International Conference, COCOA*, pages 25–34, 2014. doi:10.1007/978-3-319-12691-3_3.
- 2 Ajit Agrawal, Philip Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized steiner problem on networks. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing, STOC*, pages 134–144, 1991. doi:10.1145/103418.103437.
- 3 Afrouz Jabal Ameli, Laura Sanita, and Moritz Venzin. Learning-augmented online covering problems, 2025. doi:10.48550/arXiv.2507.06032.
- 4 Yossi Azar, Niv Buchbinder, T.-H. Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph Naor, and Debmalya Panigrahi. Online algorithms for covering and packing problems with convex objectives. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS*, pages 148–157, 2016. doi:10.1109/FOCS.2016.24.
- 5 Nikhil Bansal, Niv Buchbinder, and Joseph Naor. A primal-dual randomized algorithm for weighted paging. *J. ACM*, 59(4):19:1–19:24, 2012. doi:10.1145/2339123.2339126.
- 6 Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing problems. In *Algorithms - ESA 2005, 13th Annual European Symposium*, Lecture Notes in Computer Science, pages 689–701. Springer, 2005. doi:10.1007/11561071_61.
- 7 Niv Buchbinder and Joseph Naor. Improved bounds for online routing and packing via a primal-dual approach. In *47th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 293–304, 2006. doi:10.1109/FOCS.2006.39.
- 8 Niv Buchbinder and Joseph Naor. The design of competitive online algorithms via a primal-dual approach. *Found. Trends Theor. Comput. Sci.*, 3(2-3):93–263, 2009. doi:10.1561/04000000024.
- 9 Christian Coester, Alexa Tudose, and Alexander Turoczy. Learning-augmented online minimization with dual predictions. In *Forty-third International Conference on Machine Learning, ICML*, 2026.

- 10 William H Cunningham. On submodular function minimization. *Combinatorica*, 5(3):185–192, 1985. doi:10.1007/BF02579361.
- 11 M.S. de Lima, M.C. San Felice, and O. Lee. On generalizations of the parking permit problem and network leasing problems. *Electronic Notes in Discrete Mathematics*, 62:225–230, 2017. LAGOS’17 – IX Latin and American Algorithms, Graphs and Optimization. doi:10.1016/j.endm.2017.10.039.
- 12 Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of RANKING for online bipartite matching. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 101–107, 2013. doi:10.1137/1.9781611973105.7.
- 13 Jack Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards B*, 69(125-130):55–56, 1965.
- 14 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. doi:10.4153/CJM-1965-045-4.
- 15 Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972. doi:10.1145/321694.321699.
- 16 L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. doi:10.4153/CJM-1956-045-5.
- 17 Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995. doi:10.1137/S0097539793242618.
- 18 Dorit S. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on Computing*, 11(3):555–556, 1982. doi:10.1137/0211045.
- 19 Xinhui Hu, Arne Ludwig, Andrea Richa, and Stefan Schmid. Competitive Strategies for Online Cloud Resource Allocation with Discounts: The 2-Dimensional Parking Permit Problem. In *2015 IEEE 35th International Conference on Distributed Computing Systems (ICDCS)*, pages 93–102, 2015. doi:10.1109/ICDCS.2015.18.
- 20 Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan S. Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6):542–571, 1994. doi:10.1007/BF01189993.
- 21 Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, STOC*, pages 352–358, 1990. doi:10.1145/100216.100262.
- 22 L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13(4):383–390, January 1975. doi:10.1016/0012-365X(75)90058-8.
- 23 Christine Markarian and Friedhelm Meyer auf der Heide. Online algorithms for leasing vertex cover and leasing non-metric facility location. In *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems, ICORES*, pages 315–321, 2019. doi:10.5220/0007369503150321.
- 24 Adam Meyerson. The parking permit problem. In *46th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 274–284. IEEE Computer Society, 2005. doi:10.1109/SFCS.2005.72.
- 25 Daniel Dominic Sleator and Robert Endre Tarjan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, 1985. doi:10.1145/2786.2793.
- 26 Kharchenko Yaroslav and Kononov Alexander. A learning-augmented algorithm for the parking permit problem with three permit types. In *Mathematical Optimization Theory and Operations Research: 23rd International Conference, MOTOR*, pages 116–126, 2024. doi:10.1007/978-3-031-62792-7_8.

A

 Omitted Proofs

► **Lemma 10.** *Let S_K be defined by $S_1 := 1$ and $S_{K+1} := S_K + \frac{S_K}{\exp(S_K)-1}$. Then $S_K = \ln K + \ln \ln K + o(1)$ as $K \rightarrow \infty$.*

Proof. Let

$$h(x) := \frac{x}{e^x - 1}, \quad \Phi(x) := \int_1^x \frac{dt}{h(t)} = \int_1^x \frac{e^t - 1}{t} dt.$$

Thus the recurrence is $S_{K+1} = S_K + h(S_K)$.

First note that S_K is increasing. It cannot converge to a finite limit L , since then $h(S_K) \rightarrow h(L) > 0$, contradicting convergence. Hence $S_K \rightarrow \infty$.

Since $h(x) \rightarrow 0$ as $x \rightarrow \infty$, we further see that $S_K = o(K)$.

We now compare the discrete recurrence with the integral Φ . Since

$$\Phi'(x) = \frac{1}{h(x)} = \frac{e^x - 1}{x}$$

and

$$\Phi''(x) = \frac{(x-1)e^x + 1}{x^2} = O\left(\frac{e^x}{x}\right),$$

Taylor's formula gives, for $x \rightarrow \infty$,

$$\Phi(x + h(x)) - \Phi(x) = \Phi'(x)h(x) + O\left(\frac{e^x}{x}h(x)^2\right).$$

But $\Phi'(x)h(x) = 1$, and

$$\frac{e^x}{x}h(x)^2 = \frac{e^x}{x} \left(\frac{x}{e^x - 1}\right)^2 = O(xe^{-x}) = O(h(x)).$$

Therefore

$$\Phi(x + h(x)) - \Phi(x) = 1 + O(h(x)).$$

Applying this with $x = S_j$ and summing from $j = 1$ to $K - 1$, we obtain

$$\Phi(S_K) = K - 1 + O\left(\sum_{j=1}^{K-1} h(S_j)\right) = K - 1 + O(S_K) = K + o(K).$$

Hence $\Phi(S_K) \sim K$.

It remains to estimate Φ . By l'Hôpital's rule,

$$\lim_{x \rightarrow \infty} \frac{\Phi(x)}{e^x/x} = \lim_{x \rightarrow \infty} \frac{(e^x - 1)/x}{e^x(x-1)/x^2} = 1.$$

Thus $\Phi(x) \sim \frac{e^x}{x}$

Combining this with $\Phi(S_K) \sim K$, we get

$$K \sim \frac{e^{S_K}}{S_K}.$$

Taking logarithms yields

$$\ln K = S_K - \ln S_K + o(1). \tag{11}$$

81:16 Primal-Dual Online Algorithms for the Parking Permit Problem

Since $S_K \rightarrow \infty$, we have $\ln S_K = o(S_K)$, and therefore equation (11) implies

$$\frac{S_K}{\ln K} \rightarrow 1.$$

Consequently,

$$\ln S_K = \ln \ln K + o(1).$$

Substituting this into equation (11) proves the lemma. 