

Competitive Transaction Admission in PCNs: Online Knapsack with Positive and Negative Items

Marcin Bienkowski ✉

University of Wrocław, Poland

Julien Dallot ✉

TU Berlin, Germany

Dominik Danelski ✉

TU Berlin, Germany

Maciej Pacut ✉

TU Berlin, Germany

Stefan Schmid ✉

TU Berlin and Weizenbaum Institute, Germany

Abstract

Payment channel networks (PCNs) are a promising approach to making cryptocurrency transactions faster and more scalable. At their core, PCNs bypass the blockchain by routing transactions through intermediary channels. However, a channel can forward a transaction only if it has the necessary funds: the problem of keeping the channels balanced is a current bottleneck for the PCN’s transaction throughput.

This paper considers the problem of maximizing the number of transactions accepted by a channel in a PCN. Previous works either considered the associated optimization problem with all transactions known in advance or developed heuristics tested on particular transaction datasets. This work, however, considers the problem in its purely online form where the transactions are arbitrary and revealed one after the other.

We show that the problem can be modeled as a new online knapsack variant where the items (transaction proposals) can be either positive or negative depending on the direction of the transaction. The main contribution of this paper is a deterministic online algorithm that is $O(\log B)$ -competitive, where B is the knapsack capacity (maximum allowed channel balance). We complement this result with an asymptotically matching lower bound of $\Omega(\log B)$ which holds for any randomized algorithm, demonstrating our algorithm’s optimality.

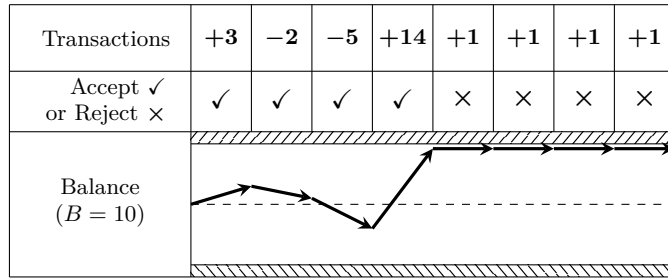
2012 ACM Subject Classification Theory of computation → Online algorithms; Networks → Network protocol design

Keywords and phrases Payment Channel Networks, Online Algorithms

Funding This project was supported by the German Research Foundation (DFG), SPP 2378 (ReNO-2), 2025–2029, and by the Polish National Science Centre grant 2022/45/B/ST6/00559

1 Introduction

Blockchain-based decentralized cryptocurrencies suffer from well-documented scalability issues [8, 7, 15] that hinder their adoption for daily payments. The promising solutions for the blockchain scalability problem are Payment Channel Networks (PCN) [20, 1], which allow users to bypass the blockchain and perform private peer-to-peer transactions. A PCN consists of a set of payment channels that allow its two end-users to perform fast, off-chain transactions to one another. The blockchain is called for an initial funding transaction at a channel’s opening and the two end-users can afterwards freely transact as long as the balance remains within the allocated funds. In general, two users willing to perform a fast transaction do not share a common channel: a typical transaction is rather routed through



■ **Figure 1** A run of the algorithm GREEDY on a single channel, for the sequence of transaction proposals shown in the first row. The second row gives the decision of GREEDY for each transaction proposal, and the third row plots the channel balance after each decision. Assuming the sequence stops after eight transaction proposals, GREEDY is suboptimal as it accepts the fourth proposal of size +14, which prevents it from accepting the last four transaction proposals of size +1. The channel balance has to be between $-B = -10$ and $B = 10$.

the network from channel to channel, each user forwarding the payment along the routing path.

A PCN crucially needs to keep its set of channels balanced to be able to route transactions with high throughput. An unbalanced channel otherwise becomes unidirectional and requires a costly call to the blockchain to regain liquidity. Efficiently balancing the channels to maximize throughput is considered the main bottleneck of PCNs and the subject of intensive research. Existing works either designed routing heuristics to minimize channel balance [3, 19, 22] or considered the associated optimization problem with full knowledge of the incoming requests [6, 21].

In this paper, we consider the problem of maximizing the transaction throughput (the number of performed transactions) of a channel in a PCN such as the Lightning network [20]. As in [21], we solve the problem locally for a single channel. Unlike previous works, however, we solve the problem in the purely online setting where the incoming transaction proposals are arbitrary and unknown to the algorithm until revealed. As we demonstrate in this paper, commonly-used greedy heuristics that accept every transaction proposal that fits in the channel are inherently suboptimal; see Figure 1 for an illustration.

In contrast, we present a deterministic online algorithm for this problem and show that it is $O(\log B)$ -competitive, where B is the maximum allowed balance of the single channel. We then show a lower bound of $\Omega(\log B)$ for all algorithms, including randomized ones, making our algorithm asymptotically optimal.

We cast the problem as a new online knapsack variant (see, e.g., [16, 17]) where the items (transaction proposals) can have positive or negative sizes (transaction amounts) depending on the direction of the transaction. The knapsack capacity corresponds to the maximum allowed balance of the single channel. Using signed items in knapsack problems is new; if we omit negative items, however, our model is similar to that of Zhou, Chakrabarty and Lukose [25] in that all our items have an (absolute) size within a known interval (as in the Lightning protocol, where channels broadcast their minimum and maximum accepted transaction amounts). However, the presence of negative items and potentially unbounded input length makes the results of [25] inapplicable to our setting.

2 Model and Notations

Knapsack and its Relation to PCN We consider an online version of the knapsack problem with items that can have both positive and negative sizes. Real-valued item sizes $\sigma_1, \sigma_2, \dots$ arrive one by one in an online fashion, and the task is to decide to either accept or reject each item when it arrives without knowledge of future items. We assume that the absolute value of any item size is from interval $[1, m]$, where m is a value known to an online algorithm.

Let B denote the *capacity* of the knapsack. At any time, we define the *state* of the knapsack as a value $s \in \mathbb{R}$ which evolves as items are accepted. Initially, the knapsack's state is $s_0 \in [-B, B]$; unless stated otherwise, we assume that $s_0 = 0$. Upon arrival of an item σ_i , the knapsack's state updates as $s \leftarrow s + \sigma_i$ in case the algorithm accepted the item, or stays unchanged otherwise.

The goal is to maximize the number of accepted items while maintaining the knapsack's state within the interval $[-B, B]$.

Modeling Payment Channel Networks Our knapsack model reflects the case of a single channel in a Payment Channel Network (PCN) such as the Lightning [20] or Raiden [1] networks. In a PCN, two users can establish a *payment channel* to perform transactions without the need for on-chain confirmations. The two end-users may perform transactions freely through the channel as long as the remaining allocated funds both remain positive. Those transactions may be for private purposes between the two end-users; however, the majority of the transactions come from outside of the channel. In that case, the transaction represents one hop as part of a routing path between distant users of the network. Over time, the PCN computes routing paths which may include the considered channel, and when this happens, the channel can either accept or reject the transaction proposal.

In our model, the sequence of items represents the transaction proposals coming from the rest of the network. The restriction on the absolute sizes of the items in $[1, m]$ models the minimum and maximum accepted transaction sizes that the channels individually decide and broadcast to the PCN (e.g., as done by the Lightning protocol [20]). Note that the minimum size of 1 implies no loss of generality as the items, the knapsack capacity and our algorithm (its acceptance curve) can be easily re-scaled to fit any positive minimum size: the parameter m can then be interpreted as the upper bound on the ratio between maximum and minimum items sizes. Note that this minimum size cannot be arbitrarily small as cryptocurrencies typically come with a smallest possible transaction amount (the Satoshi for Bitcoin and therefore the Lightning network): the channels in our dataset usually set the minimum accepted item to 1 Satoshi. Finally, our model handles the case when both end-users do not have the same initial funds by setting a non-zero initial balance s_0 .

Competitive analysis We explain here the competitive analysis framework in its maximization form; this measure is discussed at length in the book of Borodin and El-Yaniv [5]. For each input σ , let $\text{ALG}(\sigma)$ denote the total gain of the online algorithm (the number of accepted items) and $\text{OPT}(\sigma)$ denote the total gain of an optimal offline solution. A deterministic online algorithm is said to be c -competitive if there exists a parameter β , such that for any item sequence σ it holds that

$$c \cdot \text{ALG}(\sigma) \geq \text{OPT}(\sigma) - \beta. \quad (1)$$

The parameter β can depend on other model parameters such as B or m , but cannot depend on input σ . For randomized algorithms, in the definition above, we replace $\text{ALG}(\sigma)$ by its

expectation (taken over random choices of the algorithm). A competitive ratio is said to be *strict* when $\beta = 0$.

3 Contributions and Technical Novelty

Our main contribution is an asymptotically optimal competitive ratio for the transaction admission problem. Our algorithm has a competitive ratio of $O(\log B)$ (Theorem 4), where B is the knapsack capacity (channel balance has to be between $-B$ and B). For this guarantee to hold, we require a mild technical assumption on item sizes: all items must have an absolute size in $[1, m]$ where $m \leq b$ and $b = B/\ln B$. The algorithm is simple, efficient, and uses no memory: the algorithm accepts or rejects each incoming proposal solely by comparing its amount to a carefully chosen threshold depending on the current balance (see Algorithm 1 and its illustration in Figure 2).

We complement our result with a lower bound of $\Omega(\log B)$ (Theorem 8), proving the asymptotic optimality of our algorithm. Note that our lower bound is formulated as $\Omega(\log m)$ where m is the largest absolute size of an item; this bound is matching, for instance, when $m = b = B/\ln B$ (as allowed by our algorithm). This lower bound applies even to randomized algorithms, making our (deterministic) algorithm optimal across the standard frameworks of competitive analysis.

From a theoretical perspective, our results improve on the state-of-the-art of the classic online knapsack by generalizing it to signed item sizes. This generalization deeply affects the problem as its input length becomes inherently unbounded, which requires a new set of tools.

In addition, we show a worst-case input sequence for the widely-used greedy algorithm [19, 24] which shows an exponential gap between the competitive ratio of our algorithm and that of greedy (Theorem 1).

Finally, our algorithm shows very good performance in simulations (see Section 8). We show that its performance is equivalent to the commonly-used greedy algorithm, both on a single, isolated channel, and in snapshot topologies of the Lightning network against daily, non-adversarial transactions. Our theoretical results, however, ensure optimal performance guarantees even in adversarial environments, where the greedy algorithm performs poorly.

4 Lower Bound on Greedy

GREEDY [19, 24] is a widely-used algorithm that accepts every incoming item that fits in the knapsack. This section establishes that the greedy algorithm has a competitive ratio of $\Omega(m)$. Recall that B is the size of the knapsack, that is, the knapsack's state must stay in the interval $[-B, B]$, and $m \in [1, B]$ is the upper bound on the absolute item size.

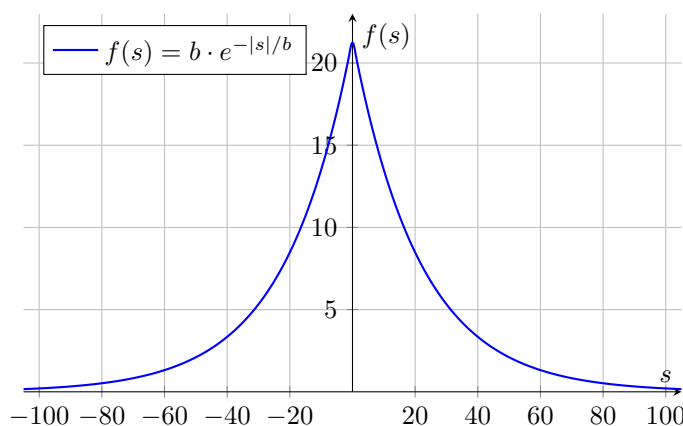
► **Theorem 1.** *The competitive ratio of GREEDY is $\Omega(m)$.*

Proof. We start with an initial knapsack's state $s_0 = 0$. Let $d = \lceil B/m \rceil$ and $m' = B/d$. Note that $m' \in [1, m]$ and $m' = \Omega(m)$. We will show a lower bound of m' on the competitive ratio of GREEDY.

We construct an input sequence σ in successive phases.

- **Phase 0.** Issue d items of size m' followed by a sequence of B items of size 1.
- **Odd phases.** Issue $2d$ items of size $-m'$ followed by a sequence of $2B$ items of size -1 .
- **Even phases.** Issue $2d$ items of size m' followed by a sequence of $2B$ items of size 1.

GREEDY accepts the first d items of phase 0 and accepts the first $2d$ items of each phase afterwards. Now consider an (offline) algorithm OFF that rejects the items accepted by



■ **Figure 2** The acceptance curve of the EXP algorithm for $B = 100$. The x-axis is the knapsack state, the y-axis is the absolute size of an item. EXP accepts an item if either that item and the knapsack's state have different signs or else if the corresponding point is below the curve.

GREEDY and vice versa. OFF accepts B items during phase 0 and $2B$ items in each phase afterwards. By summing up over all phases of σ , we obtain $\text{GREEDY}(\sigma) = (B/d) \cdot \text{OFF}(\sigma) = m' \cdot \text{OFF}(\sigma) \leq m' \cdot \text{OPT}(\sigma)$.

By increasing the number of phases, we can ensure that $\text{OPT}(\sigma)$ is arbitrarily large. This implies that the lower bound of m' holds even if we allow an arbitrary additive constant in the competitive ratio definition (1). ◀

5 Deterministic Online Algorithm

In this section, we present an online deterministic algorithm that is $O(\log B)$ -competitive when $B \geq 4.1$ and $m \leq b$ (where $b = B/\log B$).

► **Algorithm 1** (EXP). We define the function $f(x) = b \cdot \exp(-|x|/b)$. Let σ_i be the incoming item and s be the current knapsack state. Accept σ_i if σ_i and s have opposite signs or $|\sigma_i| \leq f(s)$.

Correctness We first discuss the correctness of Algorithm 1.

► **Theorem 2.** Algorithm 1 maintains the knapsack's state within $[-B, B]$ when $B \geq 4.1$.

Proof. Let $s \in [0, B]$ be a non-negative knapsack's state; the non-positive case is symmetric. Let x be the item that EXP accepts. We will show that $s + x \in [-B, B]$.

If $x < 0$, the new state is $s + x \geq s - m \geq s - b \geq s - B \geq -B$.

If $x > 0$, we define $\hat{s} = b \cdot \ln b$. Note that $f(\hat{s}) = 1$. Hence, we must have $s \leq \hat{s}$ as otherwise any positive item (of size at least 1) would be rejected by EXP. The new state is then

$$\begin{aligned}
 s + x &\leq s + f(s) && \text{(by the definition of Algorithm 1)} \\
 &\leq \hat{s} + f(\hat{s}) && \text{(as } x \mapsto x + f(x) \text{ is increasing on } [0, B]) \\
 &= b \cdot \ln b + 1 \\
 &= B - B \cdot \ln \ln B / \ln B + 1.
 \end{aligned}$$

For $B \geq 4.1$, the last quantity is at most B , concluding the proof. ◀

Competitive ratio Now that we have established correctness, we prove the performance guarantees of [Algorithm 1](#). We start with the following technical bound.

► **Lemma 3.** *Suppose EXP is in a state $s \geq 0$ and accepts item of size $x > 0$. Then,*

$$\frac{1}{f(s+x)} - \frac{1}{f(s)} \leq \frac{e-1}{b}$$

Proof. Recall that by our assumption $x \leq m \leq b$. Thus, we may use the inequality $e^y \leq 1 + (e-1) \cdot y$ holding for any $y \in [0, 1]$, obtaining

$$\begin{aligned} \exp(x/b) - 1 &\leq (e-1) \cdot (x/b) \\ &\leq (e-1) \cdot (f(s)/b) && \text{(as EXP accepts } x \text{ in state } s) \\ &= (e-1) \cdot \exp(-s/b). \end{aligned}$$

We can now estimate the desired quantity as

$$\begin{aligned} \frac{1}{f(s+x)} - \frac{1}{f(s)} &= \frac{1}{b} \cdot \left(\exp\left(\frac{s+x}{b}\right) - \exp\left(\frac{s}{b}\right) \right) \\ &= \frac{1}{b} \cdot \exp\left(\frac{s}{b}\right) \cdot \left(\exp\left(\frac{x}{b}\right) - 1 \right) \\ &\leq \frac{e-1}{b}. \end{aligned} \quad \blacktriangleleft$$

► **Theorem 4.** *EXP is $O(\log B)$ -competitive.*

Proof. Let $\sigma = \sigma_1, \sigma_2, \dots$ be an input sequence. We will prove the claim with the potential function technique. Let OPT be an optimal offline algorithm. Let s_i and s_i^* be the states of EXP and OPT, respectively, after they handle item σ_i .

Let $d_i = 2B - s_i^*$ if $s_i \geq 0$ and $d_i = 2B + s_i^*$ otherwise. Intuitively, d_i equals B plus the distance from OPT's state to that knapsack boundary which is nearest to EXP's state.

We define the potential function Φ after serving item σ_i as

$$\Phi(i) = d_i / f(s_i).$$

Note that $\Phi(i) \geq 0$. Moreover, for any i , we have $|d_i| = O(B)$ and $f(s_i) = b \cdot \exp(-|s_i|/b) \geq b \cdot \exp(-B/b) = 1/\ln B$, and thus $\Phi(i) = O(B \cdot \log B)$.

For each i , we use $\text{EXP}(i)$ and $\text{OPT}(i)$ to denote the profits of EXP and OPT on item σ_i , respectively. Our goal now is to show that the following inequality holds for each i .

$$\text{OPT}(i) + \Phi(i) - \Phi(i-1) \leq (1 + (5e-3) \cdot \ln B) \cdot \text{EXP}(i). \quad (2)$$

To show it, we first observe that flipping the signs of σ_i , s_{i-1} and s_{i-1}^* leaves $\Phi(i-1)$ and $\Phi(i)$ unchanged and keeps the same acceptance decisions for EXP regarding σ_i . We can therefore assume, without loss of generality, that $s_{i-1} \geq 0$. For succinctness, we use $\Delta_i \Phi = \Phi(i) - \Phi(i-1)$. We distinguish between two cases depending on whether EXP accepted or rejected σ_i .

1. EXP rejects σ_i .

If OPT also rejects σ_i , the inequality (2) trivially holds as both sides are zero. Thus, in the remaining part of this case, we assume that OPT accepts σ_i .

Recall that $s_{i-1} \geq 0$. The item rejection by EXP is possible only if $\sigma_i > f(s_{i-1})$. In the considered case, $s_i = s_{i-1}$. As OPT accepts σ_i , we have $s_i^* = s_{i-1}^* + \sigma_i$, and thus $d_i = d_{i-1} - \sigma_i$. Using these relations, we can estimate $\Delta_i \Phi$ as

$$\Delta_i \Phi = \frac{d_i}{f(s_i)} - \frac{d_{i-1}}{f(s_{i-1})} = \frac{d_{i-1} - \sigma_i}{f(s_i)} - \frac{d_{i-1}}{f(s_{i-1})} = -\frac{\sigma_i}{f(s_{i-1})} < -1.$$

As $\text{OPT}(i) = 1$ and $\text{EXP}(i) = 0$, (2) follows.

2. EXP accepts σ_i .

In this case $\text{EXP}(i) = 1$ and $\text{OPT}(i) \leq 1$. Hence, for showing (2), it suffices to show that $\Delta_i \Phi \leq (5e - 3) \cdot \ln B$. Using $s_i = s_{i-1} + \sigma_i$, we obtain

$$\begin{aligned} \Delta_i \Phi &= \frac{d_i}{f(s_i)} - \frac{d_{i-1}}{f(s_{i-1})} = \left(\frac{d_i}{f(s_i)} - \frac{d_i}{f(s_{i-1})} \right) + \left(\frac{d_i}{f(s_{i-1})} - \frac{d_{i-1}}{f(s_{i-1})} \right) \\ &\leq d_i \cdot \left(\frac{1}{f(s_{i-1} + \sigma_i)} - \frac{1}{f(s_{i-1})} \right) + \frac{|d_i - d_{i-1}|}{f(s_{i-1})} \end{aligned}$$

We denote the two summands in the line above as $\Delta_i^s \Phi$ and $\Delta_i^d \Phi$, respectively. We will show that $\Delta_i^s \Phi \leq (3e - 3) \cdot \ln B$ and $\Delta_i^d \Phi \leq 2e \cdot \ln B$, which will conclude the proof of this case.

- First, we upper-bound $\Delta_i^s \Phi$. If $\sigma_i < 0$, then $\Delta_i^s \Phi \leq 0$. Otherwise $\sigma_i > 0$, and we can use Lemma 3 to obtain

$$\begin{aligned} \Delta_i^s \Phi &\leq d_i \cdot \frac{e - 1}{b} \\ &\leq 3 \cdot (e - 1) \cdot \ln B. \end{aligned} \quad (\text{as } d_i \leq 2 \cdot B + s_i^* \leq 3 \cdot B)$$

- Next, we upper-bound $\Delta_i^d \Phi$. Note that the state of OPT changes at most by σ_i , and thus $|s_i^* - s_{i-1}^*| \leq \sigma_i$. We consider two sub-cases, depending on whether EXP changes the sign of its state or not.

- If EXP does not change the sign of its state, then $|d_i - d_{i-1}| = |s_i^* - s_{i-1}^*| \leq \sigma_i$. As EXP accepts σ_i , we have $\sigma_i \leq f(s_{i-1})$, and therefore $\Delta_i^d \Phi \leq 1$.
- If EXP changes the sign of its state, then $|d_i - d_{i-1}| \leq |s_i^*| + |s_{i-1}^*| \leq 2B$. However, as $\sigma_i \leq m \leq b$, the sign change is possible only if $|s_{i-1}| \leq b$, in which case $f(s_{i-1}) = b \cdot \exp(-|s_{i-1}|/b) \geq b \cdot e^{-1}$. Combining the above bounds, we get $\Delta_i^d \leq 2B/(b \cdot e^{-1}) = 2e \cdot \ln B$

Putting everything together, we have shown that $\Delta_i \Phi \leq \Delta_i^s \Phi + \Delta_i^d \leq (5e - 3) \cdot \ln B$, which concludes the proof of (2) of this case.

We proved that (2) holds for each request σ_i . By summing this relation over all requests in the input sequence, we obtain that

$$\begin{aligned} (1 + (5e - 3) \cdot \ln B) \cdot \text{EXP}(\sigma) &\geq \text{OPT}(\sigma) + \Phi(|\sigma|) - \Phi(0) \\ &\geq \text{OPT}(\sigma) - O(B \cdot \log B), \end{aligned}$$

where the last inequality follows as $\Phi(i) \geq 0$ and $\Phi(i) = O(B \log B)$ for any i . This shows that the competitive ratio of EXP is at most $(1 + (5e - 3) \cdot \ln B) = O(\log B)$. ◀

6 Randomized Lower Bound

In this section, we present a lower bound of $\Omega(\log m)$ on the competitive ratio of any randomized algorithm; recall that m is the maximum item size (in absolute value).

We may assume that $m \geq 4$, as otherwise the lower bound follows trivially. Let $q = \lceil \log_2(m/2) \rceil$ and $h = \lceil 2B/2^q \rceil$. Note that $q \geq 1$ and $2^q \leq m/2 \leq B$, and thus $h \geq 2$.

To show the lower bound, we will use Yao's minimax principle [23, 5]. To this end, we show how to construct, for any positive integer n , a probability distribution π_n over inputs, such that

- $\lim_{n \rightarrow \infty} \text{OPT}(\pi_n) = \infty$ and
- $\mathbf{E}_{\sigma \sim \pi_n}[\text{OPT}(\sigma)] / \mathbf{E}_{\sigma \sim \pi_n}[\text{DET}(\sigma)] \geq q/8$ for every deterministic algorithm DET.

Then, the minimax principle implies that the competitive ratio of every *randomized* algorithm is at least $q/8 = \Omega(\log m)$.

Random inputs For an integer $z \in [1, q]$, we define the following input subsequence, called a *positive phase of duration z* . It consists of $2 \cdot h$ items of size 2^{q-1} , followed by $4 \cdot h$ items of size 2^{q-2} , then by $8 \cdot h$ items of size 2^{q-3} , and so on, up to $2^z \cdot h$ items of size 2^{q-z} . A *negative phase of duration z* is defined analogously, but with negative item sizes.

The probability distribution π_n is defined implicitly by the following random process of generating input sequences. For an integer $t \in \{1, \dots, 2 \cdot n\}$, let z_t be an independently drawn random variable, such that for every $i \in \{1, \dots, q\}$,

$$z_t = i \quad \text{with probability } 2^{-i} \cdot \frac{1}{1 - 2^{-q}}. \quad (3)$$

The random input consists of $2 \cdot n$ phases, where the duration of the t -th phase is equal to z_t . Moreover, odd-numbered phases are positive and even-numbered ones are negative.

Estimating the expected gain of OPT We start with the following key observation about phases.

► **Lemma 5.** *Suppose an algorithm state is at most 0 before a positive phase of duration z . It is possible to collect at least $B \cdot 2^{z-q-1}$ items in this phase, and end in a state of at least $B/2$.*

Proof. Observe that the total size of all items of size 2^{q-z} in the phase is $2^z \cdot h \cdot 2^{q-z} = 2^q \cdot h = \lceil 2B/2^q \rceil \cdot 2^q \geq 2B$.

Suppose an algorithm collects as many of these items as possible. Independently of its starting state, it ends the phase in a state greater than $B - 2^{q-z} \geq B - 2^{q-1} \geq B/2$. As it starts in a state of at most 0, the total size of the collected items is at least $B/2$, and thus the number of collected items is at least $(B/2)/2^{q-z} = B \cdot 2^{z-q-1}$. ◀

► **Lemma 6.** *Fix an integer n . It holds that $\mathbf{E}_{\sigma \sim \pi_n}[\text{OPT}(\sigma)] \geq B \cdot 2^{-q} \cdot q \cdot n$.*

Proof. We consider an offline algorithm OFF that in the t -th phase accepts as many items of absolute size 2^{q-z_t} as possible. The state of OFF is 0 before the first phase, and thus satisfies the condition of Lemma 5. By the iterative application of Lemma 5 (and its symmetric counterpart for negative phases), the state of OFF is at most 0 before each positive phase, and at least 0 before each negative one. Thus, again by Lemma 5, in each phase t , OFF collects at least $B \cdot 2^{z_t-q-1}$ items. Since $\text{OPT}(\sigma) \geq \text{OFF}(\sigma)$, we have

$$\text{OPT}(\sigma) \geq B \cdot 2^{-q-1} \cdot \sum_{t=1}^{2n} 2^{z_t}.$$

Using the distribution of z_t yields $\mathbf{E}[2^{z_t}] = \sum_{i=1}^q \Pr[z_t = i] \cdot 2^i = q/(1 - 2^{-q}) > q$, and thus

$$\mathbf{E}_{\sigma \sim \pi_n}[\text{OPT}(\sigma)] \geq (B \cdot 2^{-q-1}) \cdot \sum_{t=1}^{2n} \mathbf{E}[2^{z_t}] > B \cdot 2^{-q} \cdot q \cdot n. \quad \blacktriangleleft$$

Estimating the expected gain of an online algorithm In a phase of duration z , the last batch of items (those of size 2^{q-z}) constitutes more than half of the items in that phase, and OPT can focus solely on collecting these items. An online algorithm does not know the phase duration, and it cannot simulate such a strategy.

For an input σ chosen according to the above random process, we use σ^t to denote the t -th phase of σ .

► **Lemma 7.** *Fix $t \in \{1, \dots, 2n\}$. For every starting state and every deterministic strategy DET for serving the phase σ^t , it holds that $\mathbf{E}[\text{DET}(\sigma^t)] \leq B \cdot 2^{-q+1}$.*

Proof. Without loss of generality, we assume that phase t is positive (the negative case is symmetric). Recall that the phase duration z_t is an integer between 1 and q , and for each realization of z_t , σ^t is a prefix of the phase of duration q .

Let $x_1^t, x_2^t, \dots, x_q^t$ be the number of items of sizes $2^{q-1}, 2^{q-2}, \dots, 2^{q-q}$, respectively, that would be accepted by DET when $z_t = q$. Then, on the actual phase of duration z_t , DET accepts x_j items of size 2^{q-j} for $j = 1, \dots, z_t$, i.e., $\text{DET}(\sigma^t) = \sum_{j=1}^{z_t} x_j^t$. (Note that we do not assume that x_j^t are the same for every phase t). We have

$$\sum_{i=1}^q x_i^t \cdot 2^{q-i} \leq 2B. \quad (4)$$

as otherwise DET would violate the knapsack capacity when $z_t = q$. This condition holds independently of the knapsack's state of DET at the beginning of phase t .

Using the distribution of z_t yields

$$\begin{aligned} \mathbf{E}[\text{DET}(\sigma^t)] &= \sum_{i=1}^q \Pr[z_t = i] \cdot \sum_{j=1}^i x_j^t = \sum_{j=1}^q x_j^t \cdot \sum_{i=j}^q \Pr[z_t = i] \\ &= \sum_{j=1}^q x_j^t \cdot \frac{2^{-j} - 2^{-q}}{1 - 2^{-q}} < \sum_{j=1}^q x_j^t \cdot 2^{-j} \leq 2B \cdot 2^{-q}, \end{aligned}$$

where the last inequality follows from (4). ◀

► **Theorem 8.** *The competitive ratio of any randomized algorithm is $\Omega(\log m)$.*

Proof. Construct the distribution π_n as above and fix any deterministic algorithm DET. By summing the bounds of Lemma 7 over all $2n$ phases, we obtain that

$$\mathbf{E}_{\sigma \sim \pi_n}[\text{DET}(\sigma)] = \sum_{t=1}^{2n} \mathbf{E}[\text{DET}(\sigma^t)] \leq B \cdot 2^{-q+2} \cdot n,$$

and thus using Lemma 6,

$$\mathbf{E}_{\sigma \sim \pi_n}[\text{OPT}(\sigma)] / \mathbf{E}_{\sigma \sim \pi_n}[\text{DET}(\sigma)] \geq q/4.$$

Thus, by the application of the minimax principle [23, 5], the competitive ratio of any randomized algorithm is at least $q/4 = \Omega(\log m)$.

Note that $\lim_{n \rightarrow \infty} \mathbf{E}_{\sigma \sim \pi_n}[\text{OPT}(\sigma)] = \infty$ by Lemma 6, and thus the bound holds even if we allow an arbitrary additive constant in the competitive ratio definition (1). ◀

7 Related Work

This section reviews relevant related works on channel balancing in PCNs and online knapsack problems. To our knowledge, the online knapsack problem with negative items has never been studied.

Channel balancing in PCNs Payment channel networks such as the Lightning or Raiden Networks [20, 1] enable fast, off-chain transactions between users and have emerged as a promising scaling solution for cryptocurrencies [22]. These networks come with a distinctive set of algorithmic challenges that differ from traditional networking problems. In particular, the challenges include incentivizing charging mechanisms [11], routing for re-balance maximization [3, 6] or channel depletion reduction [19, 22], and privacy concerns [24].

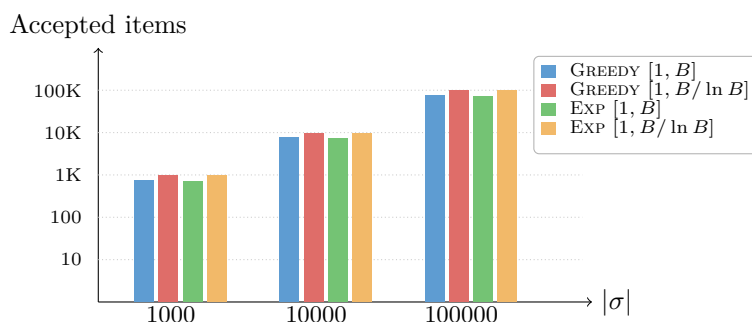
The work closest to ours is by Schmid, Svoboda and Yeo [21], who studied the offline problem of maximizing a user’s revenue in a payment channel network. The user’s possible actions are the same as ours (accept or reject an incoming transaction), but the objective is to maximize the user’s revenue. That is, an algorithm needs to strike a trade-off between charging fees (linear with the transaction size) and the channel balance which may prevent the user from accepting future transactions. Their offline approximation scheme has the ratio arbitrarily close to $1 + \sqrt{3}$.

Online knapsack in unit-profit model The online variant of the knapsack problem is a classic optimization problem that has been extensively studied for decades, with the first results dating back to the 1980s. The seminal papers by Marchetti-Spaccamela and Vercellis [17] and Lueker [16] observed that no online deterministic algorithm can achieve a bounded competitive ratio, and thus focused on the study of random instances of the problem. The negative result for deterministic algorithms holds even in the unit-profit model studied in this paper. This impossibility result is derived by combining one big item of size B , potentially followed by B/ϵ small items of size ϵ , for a fixed $\epsilon > 0$. After the first item, the adversary either stops the sequence there if the online algorithm did not accept the big item, or requests the small items. The competitive ratio can be made arbitrarily large by setting ϵ small enough. This simple negative result extends to multiple versions of the problem; see [9] for a recapitulation table.

Bounded item sizes The lower bound construction presented above can be adapted also to the case where all items have size at least 1: in this variant, no algorithm can be better than B -competitive. The situation improves if we impose an additional upper bound of $m < B$ on all item sizes. For this setting (and still assuming unit profits), Zhou, Chakrabarty and Lukose [25] gave a deterministic algorithm with competitive ratio of $O((\log B)/(1 - m/B))$. In particular, their ratio becomes $O(\log B)$ when $m = o(B)$ as is the case in our paper. They also show an asymptotically matching lower bound of $\Omega(\log B)$ holding even for randomized algorithms.

While our algorithm also achieves ratios logarithmic in B and is similar in structure to theirs, the results are not directly comparable. Our problem formulation assumes mixed-sign items, which introduces additional challenges in the analysis. One of them is that now the problem admits input sequences of unbounded length, leading to potentially unbounded profits. In particular, our lower bound applies even if we allow an arbitrary additive constant in the competitive ratio definition (1), and thus it is not subsumed by their lower bound of $\Omega(\log B)$ which only holds for the (strict) competitive ratio definition without such additive constant.

The online knapsack with bounded items is a special case of online packing problems [?, ?] that were solved using primal-dual approaches. However, the primal-dual framework does not seem to apply here, as the presence of negative items makes the problem non-monotone.



■ **Figure 3** Number of accepted items by four algorithms depending on the number of transactions, looking at one isolated channel.

Alternative models Online knapsack problems have been studied in multiple other models. In particular, a natural assumption is that profits are equal to item sizes or completely uncorrelated with them. For the former (proportional) model, deterministic algorithms still have unbounded competitive ratios, but there are $O(1)$ -competitive randomized algorithms [4].

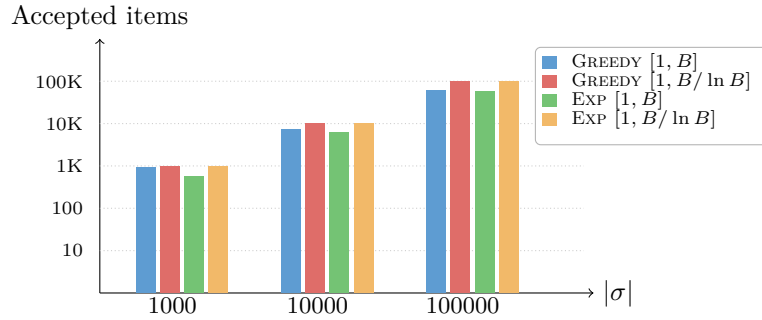
Multiple relaxations and variations have been studied to overcome the existing lower bounds. In particular, *resource augmentation* [?, ?] gives the online algorithm an advantage by allowing it to have a larger knapsack capacity than the optimal offline one. *Removable items* [?, 9, 13, 14, ?] allow the algorithm to accept an item and remove it later. *Multiple knapsacks* [?, ?, 4, ?, 9, 14] give the algorithm multiple knapsacks to fill with the goal of maximizing the total profit across all knapsacks. Finally, *random-order model* [2, ?, 12, ?, ?] assumes that the items are generated adversarially, but presented to an online algorithm in random order.

8 Evaluations

8.1 Overview of the Experiments

In this section, we evaluate [Algorithm 1](#) (EXP) and compare its performance against the commonly-used greedy algorithm [3]. The evaluation of each algorithm covers two versions: one where the absolute values of the transaction amounts are restricted to $[1, B/\ln B]$ as needed for our theoretical guarantee to hold, and one where the absolute values are in the interval $[1, B]$. We evaluate those four variants in two settings:

1. We first consider a network consisting of a single channel ([Figure 3](#)). We simulate the performance of the considered algorithms through that channel on transaction amounts generated uniformly at random within the ranges allowed by the considered model.
2. We then consider snapshot topologies of the Lightning network ([Figure 4](#)). For each transaction, we choose the two transacting nodes uniformly at random and try to transfer the transaction using a shortest path. If all the channels along the routing path accept the transaction proposal, then the transaction goes through each channel, otherwise it is rejected. The transaction amounts are generated uniformly at random within the ranges allowed by the considered model.



■ **Figure 4** Number of accepted items of four algorithms in a topology of the Lightning Network, depending on the number of transactions.

8.2 Methodology

We simulate value transfers in graphs where nodes represent users and edges represent payment channels. Any node can request a transfer to any other node. The initial balance is always 0.

Our model is implemented on top of the Python NetworkX kit. For each request, two nodes are chosen uniformly at random as the source and the destination. Then, a request size is chosen uniformly at random in a range $[1, B]$ or $[1, B/\ln B]$ depending on the considered model. Here, B is an upper bound dictated by the hash time locked contract (HTLC) request size policy and states of the channels adjacent to the source (since it knows the current balance of its own channels), and the HTLC request size policy and channel capacities of the destination (since the source can only know the predefined properties of third party channels). The minimum of these values is used. A path containing the least number of edges is searched between the source and the destination. Only channels whose capacity and HTLC policy do not limit the tested request size are taken into consideration for computing the routing path. However, this does not account for their state, as the sending party cannot know them. If a path is found, we test it with the acceptance policy under evaluation: GREEDY or EXP. If no such path exists, we try another transaction amount. If all channels on the path accept the transfer, it is counted as successful.

8.3 Curating the Dataset

For the actual Lightning Network topology, we use the Lightning Network Gossip dataset [10]. It is a collection of gossip messages (announcements of channel creation and updates) collected by the c-lightning observer node between 2018 and 2023. Using the provided time machine tool and the newest dataset provided at the time of writing (gossip-20230924), we recreate a view of the network from 23.09.2023. For it to be usable in our simulations, we augment the edge (channel) data with the capacity information that is missing in the original dataset. To this end, we use Mempool’s public REST API [18]. Finally, to remove the snapshot’s peculiarities like directed and duplicated edges, we removed multi-edges (channels with different SCID connecting the same pair of nodes) and merged double-edges having the same SCID, but present twice in the dataset, sometimes with different properties like HTLC policy (dictating maximum and minimum transaction size that could be routed through the node). In such a case, the maximum of HTLC minimums and the minimum of HTLC maximums are taken, potentially narrowing the channel. In the end, 6673 edges remained out of the original 7492 edges: we eliminated 23 multi-edges and 796 double-edges (only 131 out of

them were not direct copies of themselves).

8.4 Interpretation of the Results

Our results show that EXP, despite its strong properties against adversarial transaction sequences, keeps the same performance as the natural, widely-used GREEDY algorithm on common transactions, regardless of the network topology or the limitations on the transaction amounts. EXP therefore shows a good transaction throughput on daily transactions while ensuring a minimal throughput in any case.

9 Conclusions

This paper introduced and studied a fundamental model in decentralized finance to make cryptocurrencies scalable and able to process daily payments. The problem translates into an online variant of the knapsack problem in which items may have positive or negative sizes, subject to symmetric capacity constraints. We derived a deterministic online algorithm with asymptotically optimal competitive ratio for this problem, and showed that the ratio is tight even for randomized algorithms. Our simulations suggest that our algorithm, designed for worst-case scenarios, performs very well on average transaction sequences. An important direction for future work is to extend the model from a single channel to the full network setting of payment channel systems, where transactions must be routed across multiple channels with independently evolving states.

References

- 1 Raiden network, 2017. URL: <https://raiden.network/>.
- 2 Susanne Albers, Arindam Khan, and Leon Ladewig. Improved online algorithms for knapsack and GAP in the random order model. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, (APPROX/RANDOM)*, pages 22:1–22:23, 2019. doi:10.4230/LIPICS.APPROX-RANDOM.2019.22.
- 3 Nitin Awathare, Suraj, Akash, Vinay Joseph Ribeiro, and Umesh Bellur. REBAL: channel balancing for payment channel networks. In *29th Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 1–8, 2021. doi:10.1109/MASCOTS53633.2021.9614304.
- 4 Hans-Joachim Böckenhauer, Dennis Komm, Richard Královic, and Peter Rossmanith. On the advice complexity of the knapsack problem. In *10th Latin American Symp. on Theoretical Informatics (LATIN)*, pages 61–72, 2012. doi:10.1007/978-3-642-29344-3_6.
- 5 Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.
- 6 Krishnendu Chatterjee, Jan Matyáš Kristan, Stefan Schmid, Jakub Svoboda, and Michelle Yeo. Boosting payment channel network liquidity with topology optimization and transaction selection. In *39th Int. Symp. on Distributed Computing (DISC)*, pages 23:1–23:22, 2025. doi:10.4230/LIPICS.DISC.2025.23.
- 7 Anamika Chauhan, Om Prakash Malviya, Madhav Verma, and Tejinder Singh Mor. Blockchain and scalability. In *2018 IEEE Int. Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 122–128, 2018. doi:10.1109/QRS-C.2018.00034.
- 8 Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed E. Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. On scaling decentralized blockchains (a position paper). In *Int. Workshops on Financial Cryptography and Data Security (FC)*, pages 106–125, 2016. doi:10.1007/978-3-662-53357-4_8.

- 9 Marek Cygan, Lukasz Jez, and Jiri Sgall. Online knapsack revisited. *Theory Comput. Syst.*, 58(1):153–190, 2016. doi:10.1007/S00224-014-9566-4.
- 10 Christian Decker. Lightning network research; topology datasets. <https://github.com/lnresearch/topology>. Accessed: 2025-07-21. doi:10.5281/zenodo.4088530.
- 11 Felix Engemann, Henning Kopp, Frank Kargl, Florian Glaser, and Christof Weinhardt. Towards an economic analysis of routing in payment channel networks. In *Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers, (SERIAL@Middleware)*, pages 2:1–2:6, 2017. doi:10.1145/3152824.3152826.
- 12 Jeff Giliberti and Andreas Karrenbauer. Improved online algorithm for fractional knapsack in the random order model. In *19th Int. Workshop on Approximation and Online Algorithms (WAOA)*, pages 188–205, 2021. doi:10.1007/978-3-030-92702-8_12.
- 13 Xin Han, Yasushi Kawase, and Kazuhisa Makino. Randomized algorithms for online knapsack problems. *Theor. Comput. Sci.*, 562:395–405, 2015. doi:10.1016/J.TCS.2014.10.017.
- 14 Kazuo Iwama and Shiro Taketomi. Removable online knapsack problems. In *29th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 293–305, 2002. doi:10.1007/3-540-45465-9_26.
- 15 Anurag Jain, Shoeb Siddiqui, and Sujit Gujar. We might walk together, but I run faster: Network fairness and scalability in blockchains. In *20th Int. Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1539–1541, 2021. doi:10.5555/3463952.3464152.
- 16 George S. Lueker. Average-case analysis of off-line and on-line knapsack problems. In *6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 179–188. ACM/SIAM, 1995.
- 17 Alberto Marchetti-Spaccamela and Carlo Vercellis. Efficient on-line algorithms for the knapsack problem (extended abstract). In *14th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 445–456, 1987. doi:10.1007/3-540-18088-5_38.
- 18 Mempool. Mempool rest api documentation, 2025. URL: <https://mempool.space/docs/api/rest#get-lightning-nodes-channels>.
- 19 Suat Mercan, Enes Erdin, and Kemal Akkaya. Improving transaction success rate in cryptocurrency payment channel networks. *Comput. Commun.*, 166:196–207, 2021. doi:10.1016/J.COMCOM.2020.12.009.
- 20 Joseph Poon and Thaddeus Dryja. The bitcoin lightning network, 2016. URL: <https://lightning.network/lightning-network-paper.pdf>.
- 21 Stefan Schmid, Jakub Svoboda, and Michelle Yeo. Weighted packet selection for rechargeable links in cryptocurrency networks: Complexity and approximation. In *30th Int. Colloq. on Structural Information and Communication Complexity (SIROCCO)*, pages 576–594, 2023. doi:10.1007/978-3-031-32733-9_26.
- 22 Vibhaalakshmi Sivaraman, Shaileshh Bojja Venkatakrishnan, Kathleen Ruan, Parimarjan Negi, Lei Yang, Radhika Mittal, Giulia Fanti, and Mohammad Alizadeh. High throughput cryptocurrency routing in payment channel networks. In *17th USENIX Symp. on Networked Systems Design and Implementation (NSDI)*, pages 777–796, 2020.
- 23 Andrew Chi-Chih Yao. Probabilistic computation: towards a uniform measure of complexity. In *18th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 222–227, 1977. doi:10.1109/SFCS.1977.24.
- 24 Xiaoxue Zhang, Shouqian Shi, and Chen Qian. Webflow: Scalable and decentralized routing for payment channel networks with high resource utilization, 2021. arXiv:2109.11665.
- 25 Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. Budget constrained bidding in keyword auctions and online knapsack problems. In *Internet and Network Economics, 4th International Workshop (WINE)*, pages 566–576. Springer, 2008. doi:10.1007/978-3-540-92185-1_63.