SIAM J. COMPUT.
Vol. 30, No. 1, pp. 191–217

ALLOCATING BANDWIDTH FOR BURSTY CONNECTIONS*

JON KLEINBERG†, YUVAL RABANI‡, AND ÉVA TARDOS§

Abstract. In this paper, we undertake the first study of statistical multiplexing from the perspective of approximation algorithms. The basic issue underlying statistical multiplexing is the following: in high-speed networks, individual connections (i.e., communication sessions) are very *bursty*, with transmission rates that vary greatly over time. As such, the problem of packing multiple connections together on a link becomes more subtle than in the case when each connection is assumed to have a fixed demand.

We consider one of the most commonly studied models in this domain: that of two communicating nodes connected by a set of parallel edges, where the rate of each connection between them is a random variable. We consider three related problems: (1) stochastic load balancing, (2) stochastic bin-packing, and (3) stochastic knapsack. In the first problem the number of links is given and we want to minimize the expected value of the maximum load. In the other two problems the link capacity and an allowed *overflow probability* p are given, and the objective is to assign connections to links, so that the probability that the load of a link exceeds the link capacity is at most p. In bin-packing we need to assign each connection to a link using as few links as possible. In the knapsack problem each connection has a value, and we have only one link. The problem is to accept as many connections as possible.

For the stochastic load balancing problem we give an $O(1)$-approximation algorithm for arbitrary random variables. For the other two problems we have algorithms restricted to on-off sources (the most common special case studied in the statistical multiplexing literature), with a somewhat weaker range of performance guarantees.

A standard approach that has emerged for dealing with probabilistic resource requirements is the notion of *effective bandwidth*—this is a means of associating a fixed demand with a bursty connection that "represents" its distribution as closely as possible. Our approximation algorithms make use of the standard definition of effective bandwidth and also a new one that we introduce; the performance guarantees are based on new results showing that a combination of these measures can be used to provide bounds on the optimal solution.

Key words. combinatorial optimization, approximation algorithms, statistical multiplexing, effective bandwidth

AMS subject classifications. 05C85, 68R10, 68Q20

PII. S0097539797329142

1. Introduction.

Motivation and previous work. The issues of admission control and routing in high-speed networks have inspired recent analytical work on network routing and bandwidth allocation problems in several communities (e.g., [10, 1, 5]). One

*Received by the editors October 24, 1997; accepted for publication (in revised form) June 21, 1999; published electronically April 25, 2000. Research for this paper was supported in part by NSF through grant DMS 9505155. Portions of this work were performed while visiting the IBM Almaden Research Center.

http://www.siam.org/journals/sicomp/30-1/32914.html

†Cornell University, Department of Computer Science, Ithaca, NY 14853 (kleinberg@cs.cornell.edu). This author is supported in part by an Alfred P. Sloan Research Fellowship, an ONR Young Investigator Award, and NSF Faculty Early Career Development Award CCR-9701399.

‡Technion Institute, Computer Science Department, Haifa 32000, Israel (rabani@cs.technion.ac.il). Most of this work was done while visiting Cornell University. Support at Cornell was provided by ONR through grant N00014-96-1-0050. This work was also supported by grants from the fund for the promotion of sponsored research and the fund for the promotion of research at the Technion, and by a David and Ruth Moskowitz Academic Lecturship award.

§Cornell University, Department of Computer Science, Ithaca, NY 14853 (eva@cs.cornell.edu). The research of this author was supported in part by an NSF PYI award DMI-9157199, by NSF through grant DMS 9505155, and by ONR through grant N00014-96-1-0050.

line of work has been directed towards the development of *approximation algorithms* and *competitive on-line algorithms* for admission control and virtual circuit routing problems (see the survey by Plotkin [16]). The network model in this line of work represents the links of the network as edges of fixed *capacity* and connections as pairs of vertices with a fixed *bandwidth demand* between them. The algorithms and their analysis are motivated by this network flow perspective.

In fact, however, traffic in high-speed networks based on asynchronous transfer mode (ATM) and related technologies tends to be extremely *bursty*. The transmission rate of a single connection can vary greatly over time; there can be infrequent periods of very high *peak rate*, while the *average rate* is much lower.

One can try to avoid this issue by assigning each connection a demand equal to its maximum possible rate. The use of such a conservative approximation will ensure that edge capacities are never violated. But much of the strength of ATM comes from the advantage of *statistical multiplexing*—the packing of uncorrelated, bursty connections on the same link. In particular, suppose one is willing to tolerate a low rate of packet loss due to occasional violations of the link capacity. As the "peak" states of different connections coincide only very rarely, one can pack many more connections than is possible via the above worst-case approach and still maintain a very low rate of packet loss due to overflow.

Queueing theorists recently have devoted a great deal of study to the analysis of statistical multiplexing (see the book edited by Kelly, Zachary, and Zeidins [13]). Typically, this work models a single connection either as a discrete random variable X, with $\Pr[X = s]$ indicating the fraction of the time that the connection transmits at rate s, or as a finite-state Markov chain with a fixed transmission rate for each state. (A much-discussed case is when X is an *on-off source*. In our context, such a connection is equivalent to a weighted Bernoulli trial.) This line of work has concentrated primarily on the case of point-to-point transmission across a set of parallel links; this allows one to study the packing and load balancing issues that arise without the added complication of path selection in a large network.

One of the main concepts that has emerged from this work has been the development of a notion of *effective bandwidth* for bursty connections. This is based on the following natural idea. Suppose one is willing to tolerate a rate p of overflow on each link. One first assigns a number $\beta_p(X)$ to each connection (i.e., random variable) X, indicating the "effective" amount of bandwidth required by this connection. One then uses a standard packing or load balancing algorithm to assign connections to links, using the single number $\beta_p(X)$ as the *demand* of the connection X. This notion of effective bandwidth is indeed what underlies the modeling of routing problems as network flow questions.

Consensus has more or less been reached (see Kelly [12]) on a specific formula for β_p, first studied by Hui [10]: a scaled logarithm of the moments-generating function of X. One of its attractions is that packing according to $\beta_p(X)$ always provides a relatively *conservative estimate* in the following sense: If the sum of the effective bandwidths of a set of independent connections does not exceed the link capacity, then the probability that the sum of their transmission rates exceeds twice the capacity at any instant is at most p.

Problems studied in this paper. In this paper, we undertake the first study of the issues inherent in statistical multiplexing from the perspective of approximation algorithms. We are motivated primarily by the following fact: the queueing theoretical work discussed above does not attempt to prove that its methods, based on effective bandwidth, provide solutions that are *near-optimal* on all (or even on typical) instances. Indeed, researchers have recognized that claims about the power of the effective bandwidth approach depend critically on a number of fundamental assumptions about the nature of the underlying traffic (e.g., de Veciana and Walrand [18]). Thus an analysis of statistical multiplexing problems in the framework of approximation algorithms can provide tools for understanding the performance guarantees that can be attained in this domain.

We mentioned above that the model studied in this area concentrates primarily on the case of two communicating nodes connected by a set of parallel edges. Thus, the problem of assigning bursty *connections* to *edges* is equivalent to that of assigning (bursty) *items* to *bins*. As a result, we have a direct connection between the standard questions addressed in statistical multiplexing and stochastic versions of some of the classical resource allocation problems in combinatorial optimization. We design and analyze approximation algorithms for the following fundamental problems:

Stochastic load balancing. An *item* is a discrete random variable. We are given items $X_1, \ldots, X_n$. We want to assign each item to one of the bins $1, \ldots, m$ so as to minimize the *expected maximum weight* in any bin. That is, we want to minimize

$$\mathrm{E}\left[\max_i \sum_{X_j \in B_i} X_j\right],$$

where B_i is the set of items assigned to bin i.

Stochastic bin-packing. We are given items as above, and we define the *overflow probability* of a subset of these items to be the probability that their sum exceeds 1. We are also given a number $p \geq 0$. We want to determine the minimum number of bins (of capacity 1) that we need in order to pack all the items, so that the overflow probability of the items in each bin is at most p.

Stochastic knapsack. We are given $p \geq 0$ and a set of items $X_1, \ldots, X_n$, with item X_i having a *value* v_i. We want to find a subset of the items of maximum value, subject to the constraint that its overflow probability is at most p.

Thus, the above problems provide us with a very concrete setting in which to try assessing the power of various approaches to the statistical multiplexing of bursty connections. These problems are also the natural stochastic analogues of some of the central problems in the area of approximation algorithms; and hence we feel that their approximability is of basic interest.

Of course, each of these problems is NP-hard, since the versions in which each item X_i is *deterministic* (i.e., takes a single value with probability 1) correspond to the *minimum makespan*, *bin-packing*, and *knapsack* problems, respectively. However, the stochastic versions introduce considerable additional complications. For example, we show that even *given* a set of items, determining its overflow probability is $\#P$-complete (see section 2).

Moreover, we also show that simple approaches such as (i) applying Hui's definition of effective bandwidth [10] to the items, and then (ii) running a standard algorithm for the case of deterministic weights (e.g., Graham's lowest-fit makespan algorithm or first-fit for bin packing) can lead to results that are very far from optimal. Indeed, we show in section 2 that in a certain precise sense there is no "direct" use of effective bandwidth that can provide approximation results as strong as those we obtain.

1.1. Our results. This paper provides the first approximation algorithms for these load balancing and packing problems with stochastic items. Our algorithms make use of effective bandwidth, and their analysis is based on new results showing, roughly, that it is possible to define a notion of *effective bandwidth* that can be used to obtain bounds on the value of the optimum.

However, the relationships between the effective bandwidth and the optimum are quite subtle. In particular, while Hui's definition is a useful ingredient in our algorithm for the case of load balancing, we show in the cases of bin-packing and knapsack that it is *necessary* to use a definition of effective bandwidth that is *different* from the standard one. Our new effective bandwidth function β' has a number of additional properties that make its analysis particularly tractable. In particular, it was through β' that we were able to establish our basic relations between the function β and the value of the optimum for the case of load balancing.

Load balancing. Perhaps our strongest result is for the load balancing problem: we provide a constant-factor approximation algorithm for the optimum load for arbitrary random variables. With a somewhat larger constant, we can modify our algorithm to work in an *on-line* setting, in which items arrive in sequence and must be assigned to bins immediately.

Let us give some indication of the techniques underlying this algorithm. First, we mentioned above that the standard effective bandwidth β_p comes with an *upper bound* guarantee: if the sum of the effective bandwidths of a set of items is bounded by 1, then the probability that the total load of these items exceeds 2 is at most p. (This fact is due originally to Hui [10] and has been extended and generalized by Kelly [11], Elwalid and Mitra [4], and others.)

Our proof of the constant approximation ratio uses a new *lower bound* guarantee for effective bandwidth. Suppose we have a set of random variables $X_1, \ldots, X_n$, so that each X_i is a weighted Bernoulli trial taking on the values 0 and 2^{-i} for an integer $0 \le i \le \log\log p^{-1}$. We show that there is an absolute constant $C \le 7$ so that if the sum of the effective bandwidths of the X_i is at least C, then the probability that their sum exceeds 1 is *at least* p.

A number of issues must be resolved in order to use these bounds in the design and analysis of our algorithm. First, the upper bound guarantee holds only under some restricting assumptions on the item sizes, which are not necessarily valid for our input. Therefore, we have to handle *exceptional* items separately. Second, our lower bound concerns overflow probabilities, whereas our objective function is the expected maximum load in any bin. Finally, we have to use this lower bound in the setting of arbitrary random variables, despite the fact that the concrete result itself applies only to a restricted type of random variable.

Bin-packing and knapsack. In the case of the bin-packing and knapsack problems we consider primarily on-off sources. In our context, such a connection is equivalent to a weighted Bernoulli trial. Our emphasis on on-off sources is in keeping with the focus of much of the literature (see, e.g., the book [13]). With somewhat weaker performance guarantees, we can also handle the more general case of high-low sources: connections whose rates are always one of two positive values.

For the bin-packing problem with on-off items we give an algorithm that finds a solution with at most $O(\sqrt{\frac{\log p^{-1}}{\log\log p^{-1}}})B^* + O(\log p^{-1})$ bins, where B^* is the minimum possible number of bins. For the knapsack problem we provide an $O(\log p^{-1})$-approximation algorithm. We also provide constant-factor approximation algorithms for both problems, in which case one is allowed to relax either the size of the bin or the overflow probability by an arbitrary constant $\varepsilon > 0$. Our algorithm for bin-packing can be modified to work in an *on-line* setting, in which items arrive in sequence and must be assigned to bins immediately.

Our algorithms are based on a notion of effective bandwidth, but not the standard one in the literature. In particular, the guarantee provided by the standard definition is not strong enough for the bin-packing and knapsack problems: it says that if the sum of the effective bandwidths of a set of items is bounded by 1, then the probability that the total load of these items exceeds 2 is at most p. While such a guarantee is strong enough for the load balancing problem—a load of 2 is within a constant factor of a load of 1—it is inadequate for the bin-packing and knapsack problems, which fix hard limits on the size of each bin. Stronger guarantees without exceeding the link capacity were provided by Hui [10], Kelly [11], and Elwalid and Mitra [4] using large overflow buffers. We provide such stronger guarantees without resorting to overflow buffers. In particular, for items of large peak rate (the most difficult case for the standard definition β), we make use of our new effective bandwidth β' to provide the desired performance guarantee.

1.2. Connections with stochastic scheduling. Although we have so far expressed things in the context of bursty traffic in a network, our result on load balancing also resolves a natural problem in the area of *stochastic scheduling*.

There is a large literature on scheduling with stochastic requirements; the recent book on scheduling theory by Pinedo [15] gives an overview of the important results known in this area. In a stochastic scheduling problem, the job processing times are represented by random variables; typical assumptions are that these processing times are independent and identically distributed, and that the distribution is Poisson or exponential. For some of these cases, algorithms have been developed that guarantee an asymptotically optimal schedule with high probability (e.g., Weiss [19, 20]).

We can naturally view our load balancing problem as a scheduling problem on m identical machines (the bins), with a set of n stochastic jobs (the items). Since the problem contains the NP-hard deterministic version as a special case, we cannot expect to find an optimal solution. What our load balancing result provides is a constant approximation for the minimum makespan problem on m identical machines, when the processing time of each job can have an *arbitrary* distribution.

One distinction that arises in these scheduling problems is the following: must all the jobs be loaded onto their assigned machines immediately, or can we perform an assignment adaptively, learning the processing times of earlier jobs as they finish? Our model, since it is motivated by a circuit-routing application, takes the first approach. This is also the approach taken by, e.g., Lehtonen [14], who considers the special case of exponentially distributed processing times; that work left the case of general distributions—which we handle here—as an open problem.

2. Preliminary results and examples. For much of the paper, we will be discussing random variables that are *Bernoulli trials*. We say that a random variable X is a *Bernoulli trial of type* (q, s) if X takes the value s with probability q and the value 0 with probability $1 - q$.

The load balancing, bin-packing, and knapsack problems are all NP-complete even when all items are *deterministic* (i.e., they assume a single value with probability 1). As mentioned above, the introduction of stochastic items leads to new sources of intractability.

THEOREM 2.1. *Given Bernoulli trials $X_1, \ldots, X_n$, where X_i is of type (q_i, s_i), it is #P-complete to compute* $\Pr[\sum_i X_i > 1]$.

Proof. Membership in $\#P$ is easy to verify. We prove $\#P$-hardness by a reduction from the problem of counting the number of feasible solutions to a knapsack problem. That is, given numbers $y_1, \ldots, y_n$ and a bound B, we want to know how many subsets of $\{y_1, \ldots, y_n\}$ add up to at most B. We make two modifications to this problem which do not affect its tractability:

(i) We assume that $B = 1$.

(ii) We consider the complementary problem of counting the number of subsets of $\{y_1, \ldots, y_n\}$ that sum to more than B.

Thus, given $y_1, \ldots, y_n$, we create Bernoulli trials $X_1, \ldots, X_n$ such that X_i is of type $(\frac{1}{2}, y_i)$. Let $p = \Pr[\sum_i X_i > 1]$. The theorem follows from the fact that the number of subsets of $\{y_1, \ldots, y_n\}$ that sum to more than 1 is equal to $p \cdot 2^n$. ∎

The use of effective bandwidth is a major component in the design of our approximation algorithms. We now give some examples to show that no "direct" use of effective bandwidth will suffice in order to obtain the approximation guarantees presented in later sections. These examples also provide intuition for some of the issues that arise in dealing with stochastic items.

First we consider the load balancing problem. A natural approximation method one might consider here is Graham's lowest-fit algorithm applied to the expected values of the items. However, this fails to achieve a constant-factor approximation. This is a consequence of the following much more general fact. Let γ be any function from random variables to the nonnegative real numbers. If $X_1, \ldots, X_n$ are random variables, and ϕ is an assignment of them to m bins, we say that ϕ is γ-optimal if it minimizes the maximum sum of the γ-values of the items in any one bin.

THEOREM 2.2. *For every function γ as above, there exist $X_1, \ldots, X_n$ and a γ-optimal assignment ϕ of $X_1, \ldots, X_n$ to m bins such that the load of ϕ is $\Omega(\log m / \log\log m)$ times the optimum load.*

Proof. For an arbitrary function γ, we consider just two kinds of distributions: a Bernoulli trial of type $(m^{-\frac{1}{2}}, 1)$ and a Bernoulli trial of type $(1, 1)$. (This latter distribution is simply a deterministic item of weight 1.) By rescaling, assume that γ takes the value 1 on Bernoulli trials of type $(1, 1)$ and the value $am^{-\frac{1}{2}}$ on Bernoulli trials of type $(m^{-\frac{1}{2}}, 1)$. We consider two cases.

Case 1. $a \le \frac{\varepsilon \log m}{\log\log m}$ for some sufficiently small constant ε. In this case, we consider the following γ-optimal assignment: one item of type $(1, 1)$ in each of the $m - \sqrt{m}$ bins, and $\sqrt{m}/a$ items of type $(m^{-\frac{1}{2}}, 1)$ in each of the remaining $\sqrt{m}$ bins. With high probability, at least $\frac{\varepsilon \log m}{\log\log m}$ of the latter type of item will be on in the same bin, and hence the load of this assignment is $\Omega(\log m / \log\log m)$. By placing at most one item of each type in every bin, one can obtain a load of 2 for this problem.

Case 2. $a > \frac{\varepsilon \log m}{\log\log m}$. In this case, consider the following γ-optimal assignment ϕ: $C\sqrt{m}\log m$ items of type $(m^{-\frac{1}{2}}, 1)$ in each of $m - 1$ bins, for a sufficiently large constant C, and $aC\log m$ items of type $(1, 1)$ in the mth bin. Thus, the load of ϕ is at least $aC\log m$. However, with high probability, the maximum load in the first $m-1$ bins will be $\Theta(\log m)$, and hence the assignment that evenly balances the items of both types has load $O((1 + \frac{a}{m})\log m)$. This is better by a factor of $\Omega(\frac{am}{a+m})$. ∎

We now discuss a similar phenomenon in the case of bin-packing. Let us say that a packing of items into bins is *incompressible* if merging any two of its bins results in an infeasible packing. For the problem of packing deterministic items, a basic fact is that any incompressible packing is within a factor of 2 of optimal. In contrast, we can show the existence of a set of stochastic items that can be packed in only two bins, but for which there is an *incompressible* packing using $\Omega(p^{-\frac{1}{2}})$ bins.

THEOREM 2.3. *Consider a bin-packing problem with overflow probability* p. *There exist sets of weighted Bernoulli trials* S_1 *and* S_2 *with the following properties.*

(i) $|S_1| = |S_2| = \Omega(p^{-\frac{1}{2}})$.

(ii) *All the items of* S_1 *can be packed in a single bin.*

(iii) *All the items of* S_2 *can be packed in a single bin.*

(iv) *One cannot pack one item from* S_1 *and two from* S_2 *together in one bin.*

Thus there is a packing of $S_1 \cup S_2$ *in two bins, but the packing that uses* $\Omega(p^{-\frac{1}{2}})$ *and places one item from each set in each bin is incompressible.*

Proof. Let p be the given overflow probability, q a real number slightly greater than p, and ε a small constant. One can verify that the above properties hold for the following two sets of weighted Bernoulli trials: S_1 consists of $\varepsilon p^{-\frac{1}{2}}$ items of type $(q, 1 - \sqrt{p})$; S_2 consists of $\varepsilon p^{-\frac{1}{2}}$ items of type $(1, \sqrt{p})$. ∎

COROLLARY 2.4. *No algorithm which simply looks at a single "effective bandwidth" number for each item can provide an approximation ratio better than* $\Omega(p^{-\frac{1}{2}})$.

Proof. Note the behavior of any effective bandwidth function γ in the example of the above theorem. If $X \in S_1$ and $Y \in S_2$, then we have just argued that there exists a set of items whose effective bandwidths add up to $\gamma(X) + 2\gamma(Y)$ and which cannot be packed into one bin. But the entire set of items can be packed into two bins; and its total effective bandwidth is $\varepsilon p^{-\frac{1}{2}}[\gamma(X) + \gamma(Y)]$. This example also shows that the first-fit heuristic applied to a given item ordering can use a number of bins that is $\Omega(p^{-\frac{1}{2}})$ times optimal. ∎

The effective bandwidth we use. As discussed in the introduction, we will use both the standard definition of effective bandwidth β_p and a new *modified effective bandwidth* β'_p that turns out to be necessary in the case of bin-packing and is also used in proving our lower bounds on optimality for the load balancing problem. For a random variable X, one defines [10, 12]

$$\beta_p(X) = \frac{\log \mathrm{E}[p^{-X}]}{\log p^{-1}}. \tag{2.1}$$

For a Bernoulli trial X of type (q, s), we define its *modified effective bandwidth* by

$$\beta'_p(X) = \min\{s, sqp^{-s}\}. \tag{2.2}$$

For a set of random variables $\mathcal{R}$, we will use the notation $\beta_p(\mathcal{R}) = \sum_{X \in \mathcal{R}} \beta_p(X)$ and $\beta'_p(\mathcal{R}) = \sum_{X \in \mathcal{R}} \beta'_p(X)$.

We first give an inequality relating our modified effective bandwidth to the standard one. The proof follows from elementary calculus.

PROPOSITION 2.5. *For a Bernoulli trial* X, $\beta_p(X) \le \beta'_p(X)$.

Proof. First, we establish the following claim.

(A) *For* $a \ge 1$, *define* $f(x) = a^x - 1$ *and* $g(x) = xa^x \ln a$. *Then* $f(x) \le g(x)$ *for all* $x \in [0, 1]$.

We prove (A) by noting that

$$\lim_{x \to 0} \frac{f(x)}{g(x)} = 1,$$

and $f'(x) \le g'(x)$ for all $x \in [0, 1]$.

Now if X is of type (q, s), then we have

$$\beta_p(X) = \frac{\log\left(qp^{-s} + (1-q)\right)}{\log p^{-1}} = \frac{\log\left(1 + q(p^{-s} - 1)\right)}{\log p^{-1}}.$$

To prove the proposition, it is sufficient to show that $\beta_p(X) \le s$ and $\beta_p(X) \le sqp^{-s}$. The first of these statements follows by taking logarithms base p^{-1} of the inequality $qp^{-s} + (1-q) \le p^{-s}$. To show the second, note that by Taylor's inequality

$$\beta_p(X) \le \frac{q(p^{-s} - 1)}{\log p^{-1}},$$

and by fact (A)

$$\frac{q(p^{-s} - 1)}{\log p^{-1}} \le qsp^{-s}. \qquad \square$$

3. Stochastic load balancing. Let $X_1, X_2, \ldots, X_n$ be mutually independent random variables taking nonnegative real values. We shall refer to them as *items*. Let $\phi : \{1, \ldots, n\} \to \{1, \ldots, m\}$ be a function assigning each item X_i to one of m *bins*. We define the *load* of the assignment ϕ, denoted $\mathcal{L}(\phi)$, to be the expected maximum load on any bin; that is, $\mathcal{L}(\phi) = \mathrm{E}[\max_i \sum_{j \in \phi^{-1}(i)} X_j]$. We are interested in designing approximation algorithms for the problem of minimizing $\mathcal{L}(\phi)$ over all possible assignments ϕ. Note that the maximum of the expectations would be easy to approximate by simply load balancing the expectations.

3.1. The algorithm for on-off items. In this subsection we present an $O(1)$-approximation algorithm for the case of weighted Bernoulli trials; we then extend this to handle arbitrary distributions in the following subsection. For a Bernoulli trial of type (q, s), we can further assume that s is a power of 2—by reducing all item sizes to the nearest power of 2 we lose only a factor of 2 in the approximation ratio.

Our load balancing algorithm is on-line. It proceeds through iterations; in each iteration it maintains a current estimate of the optimum load, which will always be correct to within a constant factor. An iteration can end in one of two ways: the input can come to an end, or the iteration can *fail*. In the latter case, the estimate of the optimum is doubled, and a new iteration begins.

For ease of notation, the algorithm rescales all modified sizes that it sees so that the estimate in the current iteration is always equal to 1. An item X_i of type (q_i, s_i) is said to be *exceptional* if $s_i > 1$, and *normal* otherwise. Throughout the algorithm, we define $p = m^{-1}$ (recall that m is the number of bins) and C to be an absolute constant. ($C = 18$ is sufficient.) One iteration proceeds as follows; suppose that item X_i has just been presented.

(1) For each bin j, let B_j denote the set of all nonexceptional items from this iteration that have been assigned to j.
(2) If X_i is normal, then we assign it to the bin j with the smallest value of $\beta_p(B_j)$. If this would cause $\beta_p(B_j)$ to exceed C, then the iteration *fails*.
(3) Suppose X_i is exceptional. If the total *expected* size of all exceptional items seen in this iteration (including X_i) exceeds 1, then the iteration *fails*. Otherwise, X_i is assigned to an arbitrary bin.

To prove that this algorithm provides a constant-factor approximation, we show that (i) if an iteration does not fail, then the load of the resulting assignment is within a constant factor of the estimate for that iteration; and (ii) if iteration fails, then the load of any assignment must be at least a constant times the estimate for that iteration. We start with (ii).

Lower bounding the optimal solution. First we prove a lower bound on the optimal solution to the load balancing problem. This lower bound is the main new technical contribution of this part, and will be used also in analyzing the bin-packing and knapsack algorithm in the next two sections. In this subsection we state and prove the lower bound for the special case of weighted Bernoulli trials. (In section 3.2 we show how the general case follows from the special case.) Assume that $X_1, X_2, \ldots, X_n$ are independent Bernoulli trials such that X_i is of type (q_i, s_i). We will sometimes say that "item X_i is on" to refer to the event that $X_i = s_i$.

We use the following basic claim repeatedly.

CLAIM 3.1. *Let $\mathcal{E}_1, \ldots, \mathcal{E}_k$ be independent events, with* $\Pr[\mathcal{E}_i] = q_i$. *Let $\mathcal{E}'$ be the event that at least one of these events occurs. Let $q \le 1$ be a number such that $\sum_i q_i \ge q$. Then* $\Pr[\mathcal{E}'] \ge \frac{1}{2}q$.

Proof. Let $\bar{q} = \frac{1}{k}\sum_i q_i$.

$$\Pr[\mathcal{E}'] = 1 - \prod_i (1 - q_i) \ge 1 - (1 - \bar{q})^{\left(\frac{1}{\bar{q}}\sum_i q_i\right)}$$
$$\ge 1 - e^{-\sum_i q_i} \ge 1 - e^{-q} \ge q - \frac{1}{2}q^2 \ge \frac{1}{2}q. \qquad \square$$

Our key technical lower bound is in the following lemma. Here $p \in [0, 1]$ is a target probability (in this section we use $p = m^{-1}$).

LEMMA 3.2. *Let $X_1, \ldots, X_n$ be Bernoulli trials of types $(q_1, s_1), \ldots, (q_n, s_n)$, respectively, such that* $\log^{-1} p^{-1} \le s_i \le 1$ *for each i, and each s_i is an inverse power of* 2. *If $\sum_i \beta'_p(X_i) \ge 7$, then* $\Pr[\sum_i X_i \ge 1] \ge p$.

Proof. Our goal is to modify the given set of Bernoulli trials so as to obtain a new problem in which (i) the probability of the sum exceeding 1 is no greater than originally and (ii) the probability of the sum exceeding 1 is at least p.

If there is any X_i for which $\beta'_p(X_i) = s_i$, we lower q_i until $q_i = p^{s_i}$. This preserves the assumption that $\sum_i \beta'_p(X_i) \ge 7$.

Let s be an inverse power of two, and consider the set $W^{(s)}$ of items X_i for which $s_i = s$. We partition $W^{(s)}$ into sets $W_1^{(s)}, \ldots, W_{r_s}^{(s)}$ such that for all $j = 1, 2, \ldots, r_s - 1$, $2p^s \le \sum_{i | X_i \in W_j^{(s)}} q_i \le 3p^s$ and $\sum_{i | X_i \in W_{r_s}^{(s)}} q_i < 2p^s$. This can be done because $q_i \le p^s$ for all $X_i \in W^{(s)}$. We define a set $V^{(s)}$ of Bernoulli trials $Y_1^{(s)}, \ldots, Y_{r_s - 1}^{(s)}$, each of type (p^s, s). Intuitively, each $Y_j^{(s)}$ approximates well the behavior of $\sum_{X_i \in W_j^{(s)}} X_i$. In particular, we show that the former is stochastically dominated by the latter. We will prove the following:

(A) $\Pr[\sum_s \sum_j Y_j^{(s)} \ge 1] \le \Pr[\sum_i X_i \ge 1]$;

(B) $\beta'_p(\cup_s V^{(s)}) \ge 1$;

(C) $\Pr[\sum_s \sum_j Y_j^{(s)} \ge 1] \ge p$.

The claim clearly follows from (A) and (C).

To prove (A), we show that $\Pr[\sum_{X_i \in W_j^{(s)}} X_i \ge s] \ge p^s = \Pr[Y_j^{(s)} \ge s]$. The expression on the left-hand side is simply the probability that any of the items in $W_j^{(s)}$ is on; by Claim 3.1, the fact that $\sum_{i | X_i \in W_j^{(s)}} q_i \ge 2p^s$, and the fact that $p^s \le \frac{1}{2}$, this probability is at least p^s, and (A) follows.

To prove (B), notice that $\beta'_p(W^{(s)}_{r_s}) \le 2p^s s p^{-s} = 2s$, and for $1 \le j < r_s$, $\beta'_p(W^{(s)}_j) \le 3p^s s p^{-s} = 3s$. On the other hand, $\beta'_p(Y^{(s)}_j) = p^s s p^{-s} = s$. Thus $\beta'_p(V^{(s)}) \ge \frac{1}{3}(\beta'_p(W^{(s)}) - 2s)$. Hence

$$\begin{aligned}\beta'_p(\cup_s V^{(s)}) &= \sum_s \beta'_p(V^{(s)}) \ge \sum_s \frac{\beta'_p(W^{(s)}) - 2s}{3} \\ &= \frac{1}{3}\sum_s \beta'_p(W^{(s)}) - \frac{2}{3}\sum_s s \ge 1,\end{aligned}$$

where the last inequality follows from the fact that $\sum_s \beta'_p(W^{(s)}) \ge 7$, and $\sum_s s \le 2$ because s only takes on the values of inverse powers of 2.

To prove (C), recall that for all j, s, $\beta'_p(Y^{(s)}_j) = p^s s p^{-s} = s$. Now, let V denote a subset of $\cup_s V^{(s)}$ consisting of items whose sizes sum to 1. That such a set exists follows from (B) and the fact that all sizes are inverse powers of 2. Let $\{Y'_1, \ldots, Y'_\ell\}$ denote the items in V, and let $s'_1, \ldots, s'_\ell$ denote their sizes, respectively. Note that the probability that Y'_i is on is equal to $p^{s'_i}$.

The probability of the event $\sum_s \sum_j Y^{(s)}_j \ge 1$ is at least as large as the probability that all items in V are on. But this latter probability is equal to $\prod_{i=1}^{\ell} p^{s'_i} = p$. ∎

The lower bound for exceptional items follows by an argument using Claim 3.1.

LEMMA 3.3. *Let $X_1, \ldots, X_n$ be such that $L \le s_1 \le \cdots \le s_n$ and $\sum_i q_i s_i \ge L$. Then for all ϕ, we have $\mathcal{L}(\phi) \ge \frac{1}{2}L$.*

Proof. Without loss of generality, we may assume $\sum_i q_i s_i = L$. Let $q'_i = \sum_{j \ge i} q_j$. Let $\mathcal{E}_i$ denote the event that at least one item among $\{X_j\}_{j \ge i}$ is on, and let $q''_i = \Pr[\mathcal{E}_i]$. Note that because $\sum_i q_i s_i = L$ and $s_i \ge L$ for all i, we have $\sum_i q_i \le 1$ and hence $q'_i \le 1$ for all i. Thus, by Claim 3.1, $q''_i \ge \frac{1}{2}q'_i$. Write $s_0 = 0$ and $q'_{n+1} = 0$.

Observe that $\sum_i q_i s_i = \sum_i q'_i(s_i - s_{i-1})$, because each s_i is counted with a multiplier of q_i on the right-hand side.

Since $\Pr[X_i$ is on *and* not $\mathcal{E}_{i+1}] = q''_i - q''_{i+1}$, we have

$$\mathrm{E}[\max\{X_1, \ldots, X_n\}] \ge \sum_i s_i(q''_i - q''_{i+1}) = \sum_i q''_i(s_i - s_{i-1}).$$

Thus for any assignment ϕ we have

$$\begin{aligned}\mathcal{L}(\phi) \ge \mathrm{E}[\max\{X_1, \ldots, X_n\}] &\ge \sum_i q''_i(s_i - s_{i-1}) \\ &\ge \frac{1}{2}\sum_i q'_i(s_i - s_{i-1}) = \frac{1}{2}\sum_i q_i s_i = \frac{1}{2}L. \qquad ∎\end{aligned}$$

Our main lower bound for the load balancing problem is the following lemma.

LEMMA 3.4. *Suppose that for all i, s_i is an inverse nonnegative integral power of 2 (so $s_i \le 1$). Further suppose that $\sum_i \beta'_{m^{-1}}(X_i) \ge 17m$. Then, for all ϕ, $\mathcal{L}(\phi) = \Omega(1)$.*

Proof. Let ϕ be an arbitrary assignment of the items to bins. Let $B_1, \ldots, B_m$ denote the sets of items assigned to bins $1, \ldots, m$, respectively. Apply the following construction: as long as some set B'_i contains a subset$S$ with $\beta'_{m^{-1}}(S) \ge 8$, we put aside a minimal subset S with this property. Note that ${\beta'}_{m^{-1}}(S) \leq 9$ as the bandwidth of a single item of size at most 1 never exceeds 1. When we can no longer find such a subset, then the set of remaining items R has ${\beta'}_{m^{-1}}(R) \leq 8m$. Thus, this construction produces at least m subsets, such that each is assigned to a single bin by ϕ. We denote the first m of these subsets by $W_1, \ldots, W_m$.

Call a Bernoulli trial X of type (q, s) *small* if $s < 1/\log p^{-1}$. Using the fact that small items have $p^{-s} \leq 2$, we can see that the effective bandwidth ${\beta'}_p(X)$ of a small item is at most twice its expectation $\mathrm{E}[X] = qs$. Call a set W_i *dense* if the set of small items $S_i \subseteq W_i$ has ${\beta'}_{m^{-1}}(S_i) \geq 1$. If there exists a dense set W_i, then the expected size of W_i is at least $\frac{1}{2}$. Since $\mathcal{L}(\phi)$ is at least as large as the expected size of W_i, $\mathcal{L}(\phi) \geq \frac{1}{2}$ and the lemma follows.

Thus, we consider the case in which no W_i is dense. Let $W'_i \subseteq W_i$ denote the set of items in W_i which are not small. Since W_i is not dense, ${\beta'}_{m^{-1}}(W'_i) \geq 7$. By Lemma 3.2, the probability that size of W'_i exceeds 1 is at least m^{-1}. Hence the probability that *some* W'_i exceeds 1 is at least $1 - (1 - m^{-1})^m \geq 1 - e^{-1}$. Since $\mathcal{L}(\phi) \geq \mathrm{E}[\max\{W'_1, \ldots, W'_m\}]$, the lemma follows. $\square$

Recall that the algorithm maintains a current estimate. The iteration fails if the total effective bandwidth of the small and normal items in a bin would exceed a constant C (we use $C = 18$) or if the total expected size of all exceptional items seen in this iteration exceeds 1.

THEOREM 3.5. *Let W denote the set of items presented to the algorithm in an iteration that fails. For any assignment ϕ of W to a set of m bins we have $\mathcal{L}(\phi) = \Omega(1)$, where 1 is the estimate for the iteration.*

Proof. Let ϕ be an arbitrary assignment of items in W to bins. An iteration can fail in one of two ways: either because the expected total size of exceptional items exceeds 1, or because the assignment of the new item to any bin j would cause $\beta_p(B_j)$ to exceed C.

In the first case, Lemma 3.3 implies that $\mathcal{L}(\phi) \geq \frac{1}{2}$. Concerning the second case, consider the moment at which the iteration fails. We have $\sum_j \beta_p(B_j) \geq m(C-1)$ (because the new item's size, and therefore its effective bandwidth, cannot exceed 1). Recalling that $C \geq 18$, Lemma 3.4 asserts that $\mathcal{L}(\phi) = \Omega(1)$. $\square$

Upper bounding the solution obtained. The following proposition is essentially due to Hui [10], who stated it with $a = 2$ and $b = 1$. We give a short proof for the sake of completeness.

PROPOSITION 3.6 (see [10]). *Let $X_1, \ldots, X_n$ be independent random variables, and $X = \sum_i X_i$. Let $a > b$. If $\sum_i \beta_p(X_i) \leq b$, then $\Pr[X \geq a] \leq p^{a-b}$.*

Proof. First, if $\sum_i \beta_p(X_i) \leq b$, then $\sum_i \log \mathrm{E}[p^{-X_i}] \leq \log p^{-b}$ and hence $\prod_i \mathrm{E}[p^{-X_i}] \leq p^{-b}$.

Thus we have $\Pr[X \geq a] = \Pr[p^{-X} \geq p^{-a}] \leq p^a \mathrm{E}[p^{-X}] = p^a \prod_i \mathrm{E}[p^{-X_i}] \leq p^{a-b}$, where the first inequality follows from Markov's inequality, the equation from the independence of the X_i, and the last inequality from inequality above. $\square$

LEMMA 3.7. *Consider the assignment produced by any iteration of the algorithm. The load of this assignment is $O(1)$. (Recall that sizes are scaled so that 1 is the estimate for that iteration.)*

Proof. The expected size of the sum of exceptional items placed in this iteration is at most 1, so they only add at most 1 to the expected maximum load.

Let $S_j = \sum_{X_i \in B_j} X_i$. Let $x \geq 0$. As $\beta_p(B_j) \leq C$, $\Pr[S_j > x + C] \leq m^{-x}$ by Proposition 3.6. Let $S^* = \max\{S_1, \ldots, S_m\}$. We have$\Pr[S^* \geq y] \leq \sum_j \Pr[S_j \geq y]$.

Hence

$$\begin{aligned}
\mathrm{E}[S^*] &= \int_0^\infty \Pr[S^* \geq x]dx \leq C + 1 + \int_{C+1}^\infty \Pr[S^* \geq x]dx \\
&= C + 1 + \int_1^\infty \Pr[S^* \geq x + C]dx \\
&\leq C + 1 + \int_1^\infty m \cdot m^{-x} dx \\
&= C + 1 + m\frac{1}{m \ln m} = C + O(1),
\end{aligned}$$

from which the lemma follows. ∎

Since the estimates increase geometrically, a consequence of Lemma 3.7 is the following theorem.

THEOREM 3.8. *Let ϕ_A be the assignment produced by the algorithm. Then $\mathcal{L}(\phi_A) = O(1)$, where item sizes are scaled so that 1 is the estimate for the final iteration.*

Combining Theorems 3.8 and 3.5, we get our main result.

THEOREM 3.9. *The algorithm provides a constant-factor approximation to the minimum load.*

3.2. Extension to arbitrary distributions. We may assume that the only values taken on by our random variables are powers of 2. If not, other values are rounded down to a power of 2. As in the previous section, this increases our approximation guarantee by a factor of 2 at most. Call a random variable that only takes values that are powers of 2 *geometric*. By the following claim we can reduce the problem for geometric items to the problem for Bernoulli trial items, which we have already solved.

LEMMA 3.10. *Let X be a geometric random variable. Then there exists a set of independent Bernoulli trials $Y_1, \ldots, Y_k$, with $Y = \sum_i Y_i$, such that $\Pr[X = s] = \Pr[s \leq Y < 2s]$.*

Proof. Suppose that X takes the value s_i with probability q_i for $i = 1, \ldots, k$. Suppose that $s_1 > s_2 > \cdots > s_k$. We define Y_i to be of type (q_i', s_i), where

$$q_i' = \frac{q_i}{(1 - q_1 - \cdots - q_{i-1})}.$$

Notice that the events $X = s_i$, $i = 1, \ldots, k$, are mutually exclusive, and therefore q_i' is simply $\Pr[X = s_i \mid X \leq s_i]$. The set $\{q_i'\}$ is the solution to

$$\begin{aligned}
q_1 &= q_1', \\
q_2 &= (1 - q_1')q_2', \\
q_3 &= (1 - q_1')(1 - q_2')q_3', \\
&\vdots \\
q_k &= \left(\prod_{i=1}^{k-1}(1 - q_i')\right) q_k',
\end{aligned}$$

and hence

$$\Pr[\max_j Y_j = s_i] = q_i' \prod_{j=1}^{i-1}(1 - q_j') = q_i = \Pr[X = s_i].$$

As $s_i > \sum_{j>i} s_j$, the claim follows. □

The algorithm is essentially the same as before. It uses the standard definition of effective bandwidth (Equation (2.1)), which applies to any distribution. The only change arises from the fact that we must define what we mean by "exceptional" in this case. Each item X_i is now divided into an *exceptional part* $X_i \cdot \mathbf{1}_{\{X_i>1\}}$ and a *nonexceptional part* $X_i \cdot \mathbf{1}_{\{X_i \le 1\}}$. When the expected total value of all exceptional parts exceeds 1, the iteration fails; before this, exceptional parts are (necessarily) just packed together with their nonexceptional parts.

THEOREM 3.11. *The algorithm provides a constant-factor approximation to the minimum load.*

Proof. Recall that in the case of Bernoulli trials exceptional items could be packed in any bin. The upper bound argument follows as before, using Proposition 3.6 for the nonexceptional parts of the items.

The lower bound argument requires the approximation of each item by a sum of Bernoulli trials using Lemma 3.10. We replace each item X_i of a geometric random variable by the corresponding independent Bernoulli trials and apply the lower bound of the previous subsection to the resulting set of Bernoulli trials. □

4. Bin packing with stochastic on-off items. In this section we consider the *bin packing problem* with independent weighted Bernoulli trials, which we will refer to as "items." In addition we are given an allowed probability of overflow p. The problem is to pack the items into as few bins of size 1 each as possible, so that in each bin the probability that the total size of the items in the bin exceeds 1 is at most p. We assume throughout that $p \le \frac{1}{8}$; this is consistent with routing applications, where p is much smaller than this [4].

We develop approximation algorithms parameterized by a number ε, $0 < \varepsilon < \frac{1}{2}$. Our results show that a solution whose value is within a factor of $O(\varepsilon^{-1})$ to optimal can be obtained if we relax either the bin size or the overflow probability. That is, we compare the performance of our algorithm to the optimum for a slightly smaller bin size or overflow probability. Using these results we then give an approximation algorithm without relaxing either the bin size or the overflow probability. Our algorithms will be on-line, as before.

The basic outline of the method is as follows. As in the load balancing algorithm, we will classify items according to their sizes. For the case with relaxed sizes and/or probabilities, an item will be *small* if $s_i \le 1/\log_2 p^{-1}$, *large* if $s_i \ge \frac{1}{2}\varepsilon$ for the parameter ε, and *normal* otherwise. We pack using the expectation for small items, using the effective bandwidth $\beta_p(X)$ for normal items, and we develop techniques for packing large items based on our version of the effective bandwidth ${\beta'}_p(X)$. It can in fact be shown that the standard definition of effective bandwidth is not adequate for obtaining a strong enough approximation ratio.

For a large item of type (q_i, s_i), we effectively discretize its size, and work with its *effective size* $\bar{s}_i$; this is the reciprocal of the minimum number of copies of weight s_i that will overflow a bin of size 1: $\bar{s}_i^{-1} = \min\{j : \ js_i > 1\}$. Notice that $\bar{s}_i < s_i$ for all i.

An algorithm with relaxed bin size and probability. We start by describing a simpler version of the algorithm in which we relax both the bin size and the overflow probability. Each bin will contain items only of the same type (small, normal, or large). Each item is assigned a *weight*, according to which it is packed. Bins of each type can be packed according to any on-line bin-packing heuristic, applied to the weights; to be concrete, we will assume that the first-fit heuristic is being used.

Small items are given a weight equal to their expectation. A bin with small items will be packed so that its total weight does not exceed $\frac{1}{6}$. Each normal item X is assigned a weight of $\beta_p(X)$. A bin of normal items will be packed so that its total weight does not exceed ε.

The set of large items can have at most $\lceil 2\varepsilon^{-1} \rceil$ different effective sizes. They are classified into groups by the following two criteria.

(i) Each bin will only contain items of the same effective size.

(ii) We say that a large item X_i of type (q_i, s_i) and effective size $\bar{s}$ has *large probability* if $q_i \geq p^{\bar{s}}$ and *normal probability* otherwise. No bin will contain items of both large and normal probabilities.

We pack large probability items in bins so that fewer than $\frac{1}{\bar{s}}$ are in any bin. We pack normal probability items so that the sum of the probabilities of items in a bin does not exceed $p^{\bar{s}}/\bar{s}e$ where $e \approx 2.7..$ is the base of the natural logarithm. We now argue that the algorithm yields a feasible packing in bins of size $1 + \varepsilon$.

First we consider large items. If a bin contains items of effective size $\bar{s} = \frac{1}{k}$, then it will overflow if and only if at least k items are on. This implies that bins with large probability items do not overflow even if all items are on. Large items with normal probability are handled by the following lemma, which involves an analysis of our modified effective bandwidth.

LEMMA 4.1. *Let $X_1, \ldots, X_n$ be independent Bernoulli trials of types $\{(q_i, s_i)\}$, and assume that the effective size $\bar{s}_i = \bar{s}$ and $q_i \leq p^{\bar{s}}$ for all i. Let $X = \sum_i X_i$, and assume that $\sum_i q_i \leq p^{\bar{s}}/\bar{s}e$. Then* $\Pr[X \geq 1] \leq p$.

Proof. We get overflow in a bin if and only if at least k items are on, where $k = \frac{1}{\bar{s}}$. Let $\mathcal{I}$ denote the set of all items. For a set of items $S \subseteq \mathcal{I}$ of size k, the probability that all items in S are on is $\prod_{i \in S} q_i$. Thus the probability of overflow is at most

$$\sum_{S \subseteq I, |S|=k} \prod_{i \in S} q_i. \tag{4.1}$$

We claim that this formula is maximized for a given sum of probabilities $\sum_i q_i$ if all probabilities q_i are all the same. To see this, suppose that we have two items X_i, X_j with different probabilities, and consider modified items with probabilities $q_i' = q_j' = \frac{1}{2}(q_i + q_j)$. We now observe that the sum of probabilities has remained the same, but the probability of overflow is larger: the terms of (4.1) that contain 0 or 1 of the values q_i, q_j contribute in total the same as before, and terms containing both are each increased.

Assume now that all items have the same probability q. The sum of the probabilities of items is at most $p^{\bar{s}}/\bar{s}e$; hence, the number of these items is at most $p^{\bar{s}}/\bar{s}qe$. Now the probability that k items are on is bounded by

$$\binom{p^{\bar{s}}/q\bar{s}e}{1/\bar{s}} q^{\frac{1}{\bar{s}}} \leq \left(\frac{p^{\bar{s}}}{q}\right)^{\frac{1}{\bar{s}}} q^{\frac{1}{\bar{s}}} = p;$$

the inequality follows from the estimate $\binom{n}{k} \leq (\frac{en}{k})^k$. $\square$

The feasibility for small items follows easily from Chernoff bounds.

LEMMA 4.2. *If $X_1, \ldots, X_k$ be independent Bernoulli trials of types $(q_1, s_1), \ldots,$ (q_k, s_k), such that $s_i \leq \frac{1}{\log_2 p^{-1}}$, and $\sum_i \mathrm{E}[X_i] \leq \frac{1}{6}$, then* $\Pr[\sum_i X_i \geq 1] < p$.

Proof. We use Chernoff bounds to bound the probability that the sum exceeds 1. With $\mu = \frac{1}{6} \log p^{-1}$, we have $\Pr[\sum_i X_i > 1] < (e^{5/6}/6)^{6\mu} < 2^{-6\mu} = p$. $\square$

For the normal items, we apply Proposition 3.6 with $a = 1 + \varepsilon$ and $b = \varepsilon$.

We state this special case here for easy reference.

Lemma 4.3. *Let $X_1, \ldots, X_n$ be independent random variables, and $X = \sum_i X_i$. Let $\varepsilon > 0$. If $\sum_i \beta_p(X_i) \le \varepsilon$, then* $\Pr[X \ge 1 + \varepsilon] \le p$.

Theorem 4.4. *The on-line algorithm finds a packing of items in bins with the property that for each bin, the probability that the total size of the items in that bin exceeds $1 + \varepsilon$ is at most p.*

Note that large and small items are also feasible with bin size 1; it is only the normal items that require the relaxed bin size.

To prove the approximation ratio, we need to lower-bound the optimum. For small items, Chernoff bounds are sufficient; for normal items and large items of a given effective size we make use of a more careful analogue of Lemma 3.2.

Lemmas 4.7 and 4.8 will show that on large items of a given effective size the number of bins used by our algorithm is at most a constant factor away from the minimum possible. Since there are only $\lceil 2\varepsilon^{-1} \rceil$ different large effective sizes, this implies a bound of $O(\varepsilon^{-1})$ on large items. Lemma 4.6 shows that normal items with large total effective size (more than $5(1 + 2\varepsilon)$) have overflow probability more than $p^{1+3\varepsilon}$. This will imply that the number of bins used for normal items is at most an $O(\varepsilon^{-1})$ factor away from optimal. Finally, small items are again handled directly with Chernoff bounds.

Lemma 4.5. *Let $p < \frac{1}{2}$ and $X_1, \ldots, X_k$ be independent Bernoulli trials of types (q_1, s_1), ..., (q_k, s_k), such that $s_i \le \log_2 p^{-1}$. If $\sum_i \mathrm{E}[X_i] \ge 4$, then* $\Pr[\sum_i X_i > 1] >$ *p.*

Proof. We use Chernoff bounds to bound the probability that the sum exceeds 1. With $\mu = 4 \log p^{-1}$, we have

$$\Pr\left[\sum_i X_i \le 1\right] \le e^{-\frac{1}{2}\left(\frac{3}{4}\right)^2 \mu} < p,$$

and hence $\Pr[\sum_i X_i > 1] > 1 - p \ge p$. □

Next we consider normal items. In the load-balancing algorithm we proved a lower bound for effective bandwidth in Lemma 3.2; here we require a stronger version of this lemma. For later use we state the lemma with a parameter δ. Here we will use it with $\delta = 1$.

Lemma 4.6. *Let $X_1, \ldots, X_k$ be independent Bernoulli trials of types $(q_1, s_1), \ldots,$ (q_k, s_k), such that $s_i \ge \frac{1}{\log_2 p^{-1}}$, and $\sum_i \beta_p(X_i) \ge (3\delta + 2)(1 + 2\varepsilon)$; then*

$$\Pr\left[\sum_i X_i > \delta\right] > p^{\delta(1+2\varepsilon)+\varepsilon}.$$

Proof. Recall that $\beta_p(X) \le {\beta'}_p(X)$ for all Bernoulli trials; hence we have that $\sum_i {\beta'}_p(X_i) \ge (3\delta + 2)(1 + 2\varepsilon)$. Further, we will round up the size of each Bernoulli trial X_i to an integer power of $1 + \varepsilon$. Let X_i' denote the resulting rounded item, and let (q_i, s_i') denote its type. Rounding up cannot decrease the effective bandwidth, so we have that $\sum_i {\beta'}_p(X_i') \ge (3\delta + 2)(1 + 2\varepsilon)$.

Next we prove an analogue of Lemma 3.2 for the rounded items. We claim that with probability more than $p^{\delta(1+2\varepsilon)+\varepsilon}$, the total size of the rounded items exceeds $\delta(1 + \varepsilon)$. Notice that this implies that the total size of the original items exceeds δ with probability more than $p^{\delta(1+2\varepsilon)+\varepsilon}$. The proof is analogous to the proof of Lemma 3.2.

We may assume without loss of generality that $q_i \le p^{s_i}$ for all i. Now we have that

$$\sum_i q_i s_i p^{-s_i} \ge (3\delta + 2)(1 + 2\varepsilon). \tag{4.2}$$

We define the sets of items $W^{(s)}$ for each size s; partition $W^{(s)}$ into sets $W_1^{(s)}, \dots, W_{r_s}^{(s)}$ such that $2p^s \le \sum_{X_i \in W_j^{(s)}} q_i < 3p^s$ for $j = 1, \dots, r_s - 1$; and define the set $V^{(s)}$ of Bernoulli trials $Y_1^{(s)}, \dots, Y_{r_s-1}^{(s)}$, each of type (p^s, s), as in the proof of Lemma 3.2.

Next we want to argue that (i) the probability of the sum $\sum_{j,s} Y_j^{(s)}$ exceeding $\delta(1+\varepsilon)$ is no greater than the probability of $\sum_i X_i'$ exceeding $\delta(1+\varepsilon)$, and (ii) the probability of the sum $\sum_{j,s} Y_j^{(s)}$ exceeding $\delta(1+\varepsilon)$ is at least $p^{\delta(1+2\varepsilon)+\varepsilon}$.

To argue part (i) we show as before, using Claim 3.1, that $\Pr[\sum_{X_i' \in W_j^{(s)}} X_i' \ge s] \ge p^s = \Pr[Y_j^{(s)} \ge s]$. The fact that $p^s \le 1/2$ follows from the assumption that $s_i \ge 1/\log_2 p^{-1}$ for all i.

To show part (ii) we claim, using the notation from the proof of Lemma 3.2, that ${\beta'}_p(\cup_s V^{(s)}) > \delta(1+\varepsilon)$. To prove this, we note that as before we have ${\beta'}_p(V^{(s)}) > \frac{{\beta'}_p(W^{(s)}) - 2s}{3}$. Hence

$$\begin{aligned}
{\beta'}_p(\cup_s V^{(s)}) &= \sum_s {\beta'}_p(V^{(s)}) > \sum_s \frac{{\beta'}_p(W^{(s)}) - 2s}{3} = \frac{1}{3}\left(\sum_s {\beta'}_p(W^{(s)}) - 2\sum_s s\right) \\
&= \frac{1}{3}\left(\sum_s {\beta'}_p(W^{(s)}) - 2(1+\varepsilon)\right) \ge \delta(1+2\varepsilon),
\end{aligned}$$

since $\sum_s {\beta'}_p(W^{(s)}) \ge (3\delta+2)(1+2\varepsilon)$ by (4.2), and $\sum_s s \le (1+\varepsilon)$ since s only takes on values that are integer powers of $(1+\varepsilon)$ and at most ε.

We complete the proof of the lemma by showing $\Pr[\sum_s \sum_j Y_j^{(s)} > \delta(1+\varepsilon)] \ge p^{\delta(1+2\varepsilon)+\varepsilon}$. Note that for all j, s, ${\beta'}_p(Y_j^{(s)}) = p^s s p^{-s} = s$. Now, let V denote a subset of $\cup_s V^{(s)}$ consisting of items whose sizes sum to a number in $(\delta(1+2\varepsilon), \delta(1+2\varepsilon)+\varepsilon)$; such a set can be chosen as we have shown above that the sum of all sizes in $\cup_s V^{(s)}$ is at least $\delta(1+2\varepsilon)$, and all sizes are at most ε. Let $\{Y_1', \dots, Y_\ell'\}$ denote the items in V, with $s_1', \dots, s_\ell'$ denoting their sizes. Note that the probability that Y_i' is on is equal to $p^{s_i'}$.

The probability of the event

$$\sum_s \sum_j Y_j^{(s)} \ge \delta(1+2\varepsilon) > \delta(1+\varepsilon)$$

is at least as large as the probability that all items in V are on. But this latter probability is equal to

$$\prod_{i=1}^{\ell} p^{s_i'} = p^{\sum_{i=1}^{\ell} s_i'} > p^{\delta(1+2\varepsilon)+\varepsilon}. \qquad \square$$

Next we consider a group of large items of effective size s. The packing created by the algorithm is clearly optimal for items of large probability.

LEMMA 4.7. *If $\frac{1}{s}$ large probability items of effective size s are in the same bin, then the probability of overflow is more than p.*

Proof. Let $X_1, \ldots, X_s$ denote $\frac{1}{s}$ large probability items of effective size s. Note that if all $\frac{1}{s}$ items are on, then the total size exceeds the bin size 1. The probability of item i is $q_i > p^s$ for all i. The probability that all s items are on is therefore at least $\prod_i q_i > (p^s)^{\frac{1}{s}} = p$. ◻

Finally, consider large items of a given effective size and normal probability.

LEMMA 4.8. *Let $X_1, \ldots, X_k$ be independent Bernoulli trials of effective size s and probability $q_1, \ldots, q_k$, such that $s \geq \frac{1}{\log_2 p^{-1}}$; $q_i \leq p^s$ for all i, and $\sum_i q_i \geq 3p^s/s$. Then* $\Pr[\sum_i X_i > 1] > p$.

Proof. We need to argue that the probability that at least $\frac{1}{s}$ of the items are on exceeds p. We partition the set of items into sets $W_1, \ldots, W_{r+1}$ such that

$$2p^s < \sum_{X_i \in W_j} q_i \leq 3p^s$$

for $j = 1, \ldots, r$, and $r \geq \frac{1}{s}$. This is possible as $\sum_i q_i \geq 3p^s/s$ and $q_i \leq p^s$ for each i.

By the assumption that $p^s \leq \frac{1}{2}$, Claim 3.1 implies that in any set W_j for $j = 1, \ldots, r$ the probability that at least one of the items is on the set is more than p^s. Now the probability that at least one item is on in each of the first $\frac{1}{s}$ groups is more than $(p^s)^{\frac{1}{s}} = p$. This implies the lemma. ◻

Now we are ready to prove the general bound.

THEOREM 4.9. *For a parameter $\varepsilon \geq \frac{1}{\log_2 p^{-1}}$, the above on-line algorithm finds a packing of items in bins of size $1 + \varepsilon$ such that the number of bins used is at most $O(\varepsilon^{-1})$ times the minimum possible number of bins in any packing with bin size* 1 *and overflow probability at most $p^{1+3\varepsilon}$.*

Proof. We show that the number of bins used by our algorithm for small, normal, and large items is within $O(\varepsilon^{-1})$ of optimal.

First, suppose we use B bins for small items. Each bin is packed up to an expected value of at least $\frac{1}{6} - \frac{1}{\log p^{-1}}$ since packing an extra small item in the bin would exceed the expected value of $\frac{1}{6}$. It follows that the total expected value of all small items is at least $\frac{B(\log p^{-1} - 6)}{6 \log p^{-1}}$. Hence, if fewer than $\frac{B(\log p^{-1} - 6)}{24 \log p^{-1}}$ bins are used, some bin will overflow with probability exceeding p, by Lemma 4.5.

Next, suppose we use B bins for normal items. Each bin is packed up to a β_p-value of at least $\frac{1}{2}\varepsilon$ since adding a new normal item to a bin would exceed the total β_p value of ε, and each normal item has β_p value at most $\frac{1}{2}\varepsilon$. Therefore, the total β_p-value of normal items is at least $\frac{1}{2}\varepsilon B$. Hence, if fewer than $\frac{\varepsilon B}{10(1+2\varepsilon)}$ bins of size 1 are used for normal items, then Lemma 4.6 implies that some bin will overflow with probability exceeding p.

Finally, we consider large items of a given effective size. We show that we are within a constant factor of optimal on this set of items, where the constant does not depend on ε; thus, since there are only $\lceil 2\varepsilon^{-1} \rceil$ different effective sizes, our packing of large items will be within $O(\varepsilon^{-1})$ of optimal. First, Lemma 4.7 implies that for each effective size, the number of bins used for large items of large probability is optimal. Now suppose that we use B bins for large items of normal probability and a given effective size s. Then the total probability of this set of items is at least $Bp^s/2se$. Therefore, if fewer than $B/6e$ bins were used for this set of items, the items in at least one bin would have total probability more than $3p^s/s$, and by Lemma 4.8 the probability of overflow would exceed p. ◻

Algorithms with either relaxed bin size or probability. In fact, we can obtain the same approximation ratios (up to a constant factor) by only relaxing *either* the bin size or the overflow probability, but not both. Since the relaxed guarantees were only needed for normal items, the idea is to slightly "inflate" or "deflate" the size of the normal items that we present to the above algorithm and argue that we still do not lose too much in comparison to the optimum.

THEOREM 4.10. *There is a constant c such that for any parameter* $\varepsilon \geq \frac{c}{\log p^{-1}}$ *the following holds. There is an on-line polynomial time algorithm that finds a packing of items in bins of size* 1 *with overflow probability p, such that the number of bins used is at most* $O(\varepsilon^{-1})$ *times the minimum number of bins in any packing with bin sizes* 1 *and overflow probability at most* $p^{1+\varepsilon}$.

Proof. As just noted, the analysis for large and small items follows as before. The trouble with applying the previous analysis for normal items, of course, is that the packing created by the algorithm above might overflow bins of size exactly 1.

Here, we continue to use the effective bandwidth to pack normal items; however, for each normal item of type (q_i, s_i), we present the algorithm with an *inflated item* of type $(q_i, s_i(1+\varepsilon))$. We also set the threshold for large items at $\frac{1}{2}\varepsilon(1+\varepsilon)$, so that inflated items remain normal. Lemma 4.3 implies that the probability that the total sizes of the inflated items exceeds $1+\varepsilon$ is at most p; hence, the probability that the total size of the original items exceed 1 is at most p. For the lower bound on the optimum, we apply Lemma 4.6 to the inflated items, with $\delta = 1+\varepsilon$ to conclude that if the total effective bandwidth of the inflated items is sufficiently large then the probability that these items overflow a bin of size $1+\varepsilon$ is at least $p^{(1+\varepsilon)(1+2\varepsilon)+\varepsilon} \geq p^{1+5\varepsilon}$. Finally, we observe that a set of inflated items overflows a bin of size $1+\varepsilon$ if and only if the original items overflow a bin of size 1.

To get the bound claimed in the theorem, we must run the above algorithm with a parameter $\varepsilon' = \frac{1}{5}\varepsilon$. □

THEOREM 4.11. *For a parameter* $\varepsilon \geq \frac{1}{\log_2 p^{-1}}$, *there is a polynomial time algorithm that finds a packing of items in bins of size* $1+\varepsilon$ *with overflow probability p such that the number of bins used is at most* $O(\varepsilon^{-1})$ *times the minimum possible number of bins in any packing with bin sizes* 1 *and overflow probability at most p.*

Proof. We use an algorithm similar to that of Theorem 4.10, except that now we decrease the size of each normal item by a factor of $1-\varepsilon$. Lemma 4.3 implies that the decreased sized items do not overflow a bin of size $1+\varepsilon$, and hence the original items do not overflow a bin of size $\frac{1+\varepsilon}{1-\varepsilon} \leq 1+4\varepsilon$.

To obtain the lower bound, we want to prove that if the total effective bandwidth of the decreased sized items is sufficiently large, then the probability that the total size of these items exceeds $1-\varepsilon$ is at least p. This will imply that the total size of the original items is at least 1 with probability at least p. The proof follows from Lemma 4.6 applied to the decreased item sizes and $\delta = \frac{1-\varepsilon}{1+\varepsilon}$.

To get the bound claimed in the theorem, we must run the above algorithm with a parameter $\varepsilon' = \frac{1}{2}\varepsilon$. □

An algorithm without relaxing bin size or probability. In this section we use the results above to obtain an approximation algorithm without relaxing either the bin size or the capacity. In fact, our algorithm will simply be the on-line algorithm from the previous section, with $\varepsilon = \frac{1}{\log_2 p^{-1}}$. Thus, there will be no items classified as *normal*—only small and large. One can give a weak analysis of this algorithm as follows: since the relaxed probabilities and sizes were only required for normal items, this algorithm produces a packing that is with $O(\varepsilon^{-1}) = O(\log p^{-1})$ times the optimum with bin size 1 and overflow probability p.

Our goal in this section is to give a more involved analysis of the same algorithm, showing that its performance is actually much better than this: it produces a packing with $O(\sqrt{\frac{\log p^{-1}}{\log\log p^{-1}}})B^* + O(\log p^{-1})$ bins, where B^* is the optimum number of bins required (with size 1 and overflow probability p).

The main step of the analysis is the following extension of Lemma 3.2.

LEMMA 4.12. *Let* $\varepsilon = \frac{1}{6}\sqrt{\frac{\log\log p^{-1}}{\log p^{-1}}}$. *If* $X_1, \ldots, X_k$ *are independent Bernoulli trials of types* $(q_1, s_1), \ldots, (q_k, s_k)$, $\varepsilon \geq s_i \geq \frac{1}{\log_2 p^{-1}}$, *and* $\sum_i {\beta'}_p(X_i) \geq 7\varepsilon^{-1}$, *then* $\Pr[\sum_i X_i > 1] > p$.

The proof relies heavily on our modified effective bandwidth, with a grouping scheme as in the proof of Lemma 3.2. However, we cannot afford to analyze the groups in each effective size separately; thus we require a combinatorial argument which analyzes the antichain of minimal collections of groups that would cause the bin to overflow.

Before proving this lemma, we require a simple combinatorial fact. Let S be a set of size n, let $k \leq \ell \leq n$, and let $\mathcal{F}_{k\ell}$ denote the collection of all subsets of S whose size is at least k and at most ℓ. We say that $\mathcal{I} \subseteq \mathcal{F}_{k\ell}$ is an *antichain* if no set in $\mathcal{I}$ contains any other set, and a *maximal antichain* if it is maximal with this property.

CLAIM 4.13. *Assume* $k \leq \ell \leq \frac{n}{2}$. *Then the number of elements in a maximal antichain* $\mathcal{I} \subset \mathcal{F}_{k\ell}$ *is at least* $\binom{n}{k}/\binom{\ell}{k}$.

Proof. Consider a maximal antichain $\mathcal{A}$. Each k-element set S must be contained in one of the sets T of the maximal antichain $\mathcal{A}$. An antichain element T can contain up to $\binom{\ell}{k}$ k-element subsets, and there are altogether $\binom{n}{k}$ k-element sets. This implies the claim. ∎

Proof of Lemma 4.12. The proof starts out analogous to the proof of Lemma 4.6. We round item sizes up to a power of $(1+\varepsilon)$ and assume without loss of generality that all items have normal probability, i.e., $q_i \leq p^{s_i}$. Then we have that $\sum_i q_i s_i p^{-s_i} \geq 7\varepsilon^{-1}$.

We partition the set of items into sets $W_j^{(s)}$ for each size s such that $2p^s \leq \sum_{X_i \in W^j_{(s)}} q_i \leq 3p^s$ for $j = 1, \ldots, r_s$; and define the set $V^{(s)}$ of Bernoulli trials $Y_1^{(s)}, \ldots, Y_{r_s-1}^{(s)}$, each of type (p^s, s), as in the proof of Lemma 3.2. As before we have that ${\beta'}_p(V^{(s)}) \geq \frac{{\beta'}_p(W^{(s)})-2s}{3}$. Further $\sum_s s \leq 1+\varepsilon$, hence we get that ${\beta'}_p(\cup_s V^{(s)}) \geq 2\varepsilon^{-1}$. Note also that ${\beta'}_p(\cup_s V^{(s)}) \leq \frac{1}{2}{\beta'}_p(\cup_s W^{(s)}) \leq \frac{7}{2}\varepsilon^{-1}$.

Next we form groups $G_1, \ldots, G_k$ from the items in $\cup_s V^{(s)}$ so that in each group G_j for $j = 1, \ldots, k$ the sum of the sizes is in the range $[\varepsilon, 2\varepsilon)$ and $k \geq \frac{7}{2}\varepsilon^{-2}$. This is possible as each item has size at most ε, so we can form groups of the right size, and the number of groups that can be formed is at least ε^{-2}, since ${\beta'}_p(\cup_s V^{(s)}) \geq \frac{7}{2}\varepsilon^{-1}$, and the effective bandwidth of each item in $\cup_s V^{(s)}$ is equal to its size by definition (since $p^s s p^{-s} = s$). Moreover, the number of groups formed is at most $\varepsilon^{-1} \cdot {\beta'}_p(\cup_s V^{(s)}) \leq \frac{7}{2}\varepsilon^{-2}$.

A subset $\mathcal{I} \subseteq \{1, \ldots, k\}$ is called *critical* if it is minimal subset to the property that the total size of the items in $\cup_{j\in\mathcal{I}} G_i$ exceeds 1. We note the following facts:

- The number of groups in a critical set is at least $1 + \frac{\varepsilon^{-1}}{2}$ and at most $1 + \varepsilon^{-1}$.
- The probability that all items are on in the groups of a critical set is at least $p^{1+2\varepsilon}$. This follows from the facts that the total size of items in the groups of a critical set is at most $1 + 2\varepsilon$, and each item in $\cup_j G_j$ of size s has

probability p^s.

- The number of critical sets is at least $\frac{1}{2}(\frac{\varepsilon^{-1}}{2})^{-\frac{\varepsilon^{-1}}{2}}$. To see this consider the set of critical sets. The critical sets form a maximal antichain. We want to use Claim 4.13 for this antichain. From the first fact we see that the claim should be applied to $1+\frac{\varepsilon^{-1}}{2} \leq 1+\varepsilon^{-1} \leq \frac{k}{2}$. We have that $k \geq \varepsilon^{-2}$; therefore, the number of critical sets is at least

$$\binom{\varepsilon^{-2}}{1+\frac{\varepsilon^{-1}}{2}} \Big/ \binom{1+\varepsilon^{-1}}{1+\frac{\varepsilon^{-1}}{2}}.$$

To get the claimed bound above, we bound $\binom{1+\varepsilon^{-1}}{1+\frac{\varepsilon^{-1}}{2}} \leq 2^{1+\varepsilon^{-1}}$, and $\binom{\varepsilon^{-2}}{1+\frac{\varepsilon^{-1}}{2}} \geq (2\varepsilon{-}1)^{\frac{\varepsilon^{-1}}{2}}$.

We say that a group is *on* if all elements of the group are on, and the group is *off* if at least one element in the group is not on. The probability that a group is on is at least p^{ε} and at most $p^{2\varepsilon}$. Consider a critical set $\mathcal{I}$. The probability that all groups not in $\mathcal{I}$ are off is at least

$$(1-p^{\varepsilon})^k = (1-p^{\varepsilon})^{\frac{7}{2}\varepsilon^{-2}} \geq e^{-\frac{\frac{7}{2}\varepsilon^{-2}}{p^{-\varepsilon}}}.$$

By the choice of ε we have that

$$\varepsilon^{-2} = 36 \cdot \frac{\log p^{-1}}{\log\log p^{-1}}$$

and $p^{-\varepsilon} = e^{\Theta(\sqrt{\log p^{-1} \log\log p^{-1}})}$, and so $\frac{7}{2}\varepsilon^{-2} \leq p^{-\varepsilon}$ and the probability that all groups not in $\mathcal{I}$ are off is at least e^{-1}.

Now, the probability of overflow is at least the sum, over all critical sets $\mathcal{I}$, of the probability that the groups which are on are precisely those in $\mathcal{I}$. Thus, by the above bounds, we have all

$$\Pr\left[\sum_i X_i \geq 1\right] \geq \frac{1}{2}\left(\frac{\varepsilon^{-1}}{2}\right)^{-\frac{\varepsilon^{-1}}{2}} p^{1+2\varepsilon}\frac{1}{e}.$$

Finally, since $p \geq (18\varepsilon^2)^{(\varepsilon^{-2}/18)}$, it is straightforward to see that

$$\frac{1}{2}\left(\frac{\varepsilon^{-1}}{2}\right)^{-\frac{\varepsilon^{-1}}{2}} p^{2\varepsilon}\frac{1}{e} > 1. \qquad \square$$

This lemma allows us to give a stronger analysis of our algorithm: although the algorithm only recognizes small and large items, our analysis further partitions the large items depending on whether their sizes are smaller or larger than $\varepsilon = \frac{1}{6}\sqrt{\frac{\log\log p^{-1}}{\log p^{-1}}}$. For large items with sizes below ε, we apply Lemma 4.12.

THEOREM 4.14. *The above algorithm finds a packing of items in bins of size* 1 *with overflow probability* p *such that the number of bins used is at most* $O(\sqrt{\frac{\log p^{-1}}{\log\log p^{-1}}})$ $B^* + O(\log p^{-1})$, *where* B^* *is the minimum possible number of bins.*

Proof. Although the algorithm only recognizes small and large items, our analysis makes use of three types of items. Let $\varepsilon = \frac{1}{6}\sqrt{\frac{\log\log p^{-1}}{\log p^{-1}}}$, and say that an item of type

(q, s) is *large* if $s \geq \frac{1}{2}\varepsilon^{-1}$, *small* if $s \leq \log p^{-1}$, and *normal* otherwise. By earlier arguments, the algorithm is within a constant factor of optimal on small items and within an $O(\varepsilon^{-1})$ factor on large items. The problem is that the algorithm treats normal items as though they were large and packs them according to ${\beta'}_p$ applied to their effective sizes. Since there are $\Theta(\log p^{-1})$ different effective sizes of normal items, our analysis cannot consider each effective size separately. Thus, we use Lemma 4.12.

As before, we distinguish (normal) items as having *large* or *normal* probability. By giving up a factor of 2, we can still afford to analyze bins with items of normal probability separately from those with items of large probability. We say that a bin with large probability items of effective size s is *filled* if there are $\frac{1}{s} - 1$ items in the bin. For each effective size there is at most one bin of large probability items that is not filled. For a bin with normal probability items we say that the bin is *filled* if the total probability of the items in the bin is at least $p^s/2se$ (half of the maximum possible). Again, for each effective size there is at most one bin of normal probability items that is not filled. So the number of nonfilled bins of normal items is $O(\log p^{-1})$.

Next we consider filled bins. We claim that the total ${\beta'}_p$-value in a filled bin is at least $\frac{1}{2e}$. To show this, we first recall that the effective size is smaller than the real size, and that effective bandwidth is monotone in the size. This implies that a large probability item X of effective size s has effective bandwidth ${\beta'}_p(X) \geq s$. Therefore, the effective bandwidth of a bin filled with large probability items of effective size s is at least $s(\frac{1}{s} - 1) = 1 - s$. A small probability item X of probability q and effective size s has effective bandwidth at least ${\beta'}_p(X) \geq sqp^{-s}$, and so the effective bandwidth of a bin filled with normal probability items of effective size s is at least $\frac{p^s}{2es} p^{-s} s = \frac{1}{2e}$.

If our algorithm produces B filled bins, then the total effective bandwidth over all items in filled bins is at least $\frac{B}{2e}$. Lemma 4.12 implies that any packing of these items with fewer than $\frac{\varepsilon B}{14e}$ bins would result in at least one bin with too high an overflow probability. Thus the number of bins used for normal sized items is at most $O(\varepsilon^{-1})B^* + O(\log p^{-1})$, and we are finished. □

It is natural to ask whether this analysis can be further tightened to show that the same algorithm is in fact producing a packing with $O(B^*) + O(\log p^{-1})$ bins. In fact this is not possible; this is contained in the following theorem.

THEOREM 4.15. *There exist instances in which B^* is arbitrarily large, and the above algorithm uses more than $B^* \cdot \Omega(\log \log \log \log p^{-1})$ bins.*

Proof. Let b be an arbitrarily large constant, let $k = \frac{\ln \ln p^{-1}}{\ln \ln \ln p^{-1}}$, and let J be the set of all prime numbers less than or equal to k. For $r \in J$ and $1 \leq i \leq b$, let X_r^i be a Bernoulli trial of type $(p^{1/r}, 1/r)$.

First, we claim that the set of items $S^i = \{X_r^i : r \in J\}$ can be packed in a single bin. To prove this, consider any set $S' \subset S$ whose sizes sum to a number strictly greater than 1; call such a set *large*. Since the sizes of the items in S' have denominators that are pairwise relatively prime, the sum of these sizes must be at least $1 + 1/k!$. Thus we have

$$\Pr[S \text{ overflows}] \leq \sum_{large\, S' \subset S} \Pr[S' \text{ is on}] \leq 2^k p^{1+1/k!} \leq p,$$

with the last inequality following from the fact that

$$k < \frac{1}{k!} \ln p^{-1}.$$

Thus, the set of all items can be packed in b bins.

Now consider the packing produced by our algorithm. Of the items of size r, it will pack at most r in each bin. Thus, the total number of bins it produces will be at least

$$\sum_{r \in J} \frac{b}{r} = \Theta(b \log \log k) = \Theta(b \log \log \log \log p^{-1}). \qquad \square$$

The form of the final bound suggests that it is possible that our analysis could be tightened further, albeit not to provide a constant ratio.

5. The knapsack problem. Finally, we consider the knapsack problem. First we consider a simple version of the knapsack problem with items $X_1, X_2, \ldots, X_n$ that are independent Bernoulli trials. Each item has a value v_i, and we are given a knapsack size, say 1, and an allowed probability of overflow p. The problem is to find a set of items of maximum value such that the probability that the total size of the set exceeds 1 is at most p.

The lower bounds and techniques developed in the previous section yield similar results for the knapsack problem. We distinguish items by their sizes (small, normal, and large), we group large items by their effective size, and we distinguish large and small probability items just as in the previous section. The solution we construct for the knapsack problem only contains one type of item (either small, normal, or large with a given effective bandwidth). We will look for a near-optimal solution in each of these groups and select the best alternative. Thus, we can show the following.

THEOREM 5.1. *Let $X_1, \ldots, X_n$ be independent Bernoulli trials.*

- *There is a polynomial time algorithm that finds a solution to the knapsack problem with items $X_1, \ldots, X_n$ of value at least an $O(\log p^{-1})$ fraction of the optimum.*
- *For any $\varepsilon > 0$, there is a polynomial time algorithm that finds a solution to the knapsack problem, using knapsack size $1 + \varepsilon$ and overflow probability p, of value at least an $O(\varepsilon^{-1})$ fraction of the maximum possible with a knapsack of size 1 and overflow probability p.*
- *For any $\varepsilon > 0$, there is a polynomial time algorithm that finds a solution to the knapsack problem, using knapsack size 1 and overflow probability p, of value at least an $O(\varepsilon^{-1})$ fraction of the maximum possible with a knapsack of size 1 and overflow probability $p^{1+\varepsilon}$.*

Proof. For small items we use a knapsack approximation algorithm to find a set of items of approximately maximum value with at most a total of $\frac{1}{2}$ expected value.

For normal items we use a knapsack approximation algorithm to find a set of items of approximately maximum value with at most a total of ε total effective bandwidth. As in the previous section, we need to increase or decrease the item sizes by a factor of $(1 + \varepsilon)$ before computing the effective bandwidth depending on the type of result desired.

We group large items according to their effective size. For large items of effective size s we either (i) take the (at most) $\frac{1-s}{s}$ items of largest value; or (ii) use a knapsack approximation algorithm on the large items of normal probability to find a set of approximately maximum value with a total probability of at most $\frac{p^s}{es}$.

To prove the lower bound we note that the optimal value of a deterministic knapsack problem grows only linearly with the knapsack size, as long as new items of larger size are not considered. Our approximation algorithm is simply the greedy algorithm, which either (i) takes the single most valuable item, or (ii) orders the items in decreasing order of value divided by weight (i.e., expectation, effective bandwidth, or probability) and greedily fills the knapsack in this order.

FACT 5.2. *Consider a deterministic knapsack problem where the knapsack has size* 1*, and all items have size at most* 1*.*

- *The value of the solution obtained by the above greedy method is at least a* $\frac{1}{2}$ *of the optimal.*
- *For any* $c > 1$*, the value of the optimal solution with knapsacks of size* 1 *is at least a fraction of* $\frac{1}{2c}$ *of the optimal value with knapsack size* c*.*

For small items Lemma 4.2 implies that the resulting knapsack solution is feasible, and Lemmas 5.2 and 4.2 imply that its value is within a constant factor of the optimal packing of small items.

For normal items we use Lemma 4.3 to show either that (1) the packing obtained is feasible with a knapsack of size $1 + \varepsilon$ or (2) if we increased sizes before computing the effective bandwidth, then it is feasible with a knapsack of size 1. Lemmas 5.2 and 4.6 imply that the packing obtained is within an $O(\varepsilon^{-1})$ factor of any packing of normal items with either (1) a knapsack of size 1 and overflow probability at most p or (2) a knapsack of size 1 and overflow probability at most $p^{1+\varepsilon}$.

For large items of effective size s, Lemma 4.1 shows that the solution is feasible, and Lemmas 4.7 and 4.8 and Fact 5.2 imply that the solution is within a constant factor to any packing using items of effective size s.

In total we get an $O(1)$ approximation algorithm for small items, and items of a given effective size. There are $O(\varepsilon^{-1})$ different effective sizes. For normal items, we get an $O(\varepsilon^{-1})$ approximation. The approximation ratio of the best solution among these options is the sum of the approximation ratios of the special cases. Thus, the theorem follows. ∎

Extension to other distributions. Next we extend the solution to a distribution that is somewhat more general than the on-off distribution we have been considering so far. We assume that each item is parameterized by numbers lo, hi, and q, where the item is of size hi with probability q and of size $lo \leq hi$ otherwise. This kind of item is a simple model of a bursty communication, with lo being the normal rate of transmission and hi being a burst that occurs with probability q.

Consider the optimal knapsack packing. Assume we used items $X_1, \ldots, X_k$ in the packing, and let $g = \sum_i lo_i$. The idea is that we guess the value of g in the optimal solution, and for every guess we look for packings that are feasible and in which the total of the lo values is at most g.

We define small items depending on the size $s_i = hi_i - lo_i$ of the probabilistic part, limiting the value s_i to be at most half of what it was for the size in the case of Bernoulli trials. We pack small items using expectation, subject to the the fact that the total expectation is at most $\frac{1}{2}$, and we use Lemmas 4.2 and 4.5 and Fact 5.2 to see that the value of the packing is within a constant factor of the maximum possible using small items.

We define normal items also using $s_i = hi_i - lo_i$ and pack normal items using the effective bandwidth. However, to compute the right effective bandwidth for normal items we need to know $1 - g$, the amount of space left for the probabilistic part of the items, since the effective bandwidth formula assumed that the bin size is 1, so we will have to rescale the bin size to apply the formula. Notice, however, that it suffices to know $1 - g$ roughly up to a factor of $1 + \varepsilon$.

Notice that it is essentially no loss of generality to assume that $g \leq \frac{1}{2}$. Among items with lo values above $\frac{1}{2}$ at most one can be in the knapsack, so we can either pack a single one of these items in the knapsack by itself or assume that we are not using any of them. For items with $lo \le \frac{1}{2}$ the restriction that $g \le \frac{1}{2}$ will not change the optimum value by more than a small constant factor. Therefore, we need only to consider $O(\varepsilon^{-1})$ different values for g to obtain an $O(\varepsilon^{-1})$ approximate solution the optimal value of a solution using normal items.

In the case of Bernoulli trials we used the greedy method to pack items in each category into a knapsack using expectation, effective size, or the probability depending on whether we considered items that are small, normal, or large. Here we cannot use the greedy method to find a solution to the resulting deterministic problem, as we have to consider an extra parameter, the sum of the lo values. We use the following method instead.

A deterministic *two-dimensional knapsack problem* is defined by a set of items $X_1, \ldots, X_n$, each with a value v_i, size s_i, and weight w_i. In addition, we are given a knapsack size S and a weight limit W. The problem is to find a subset $\mathcal{I}$ of items of maximum total value so that the total size $\sum_{i \in \mathcal{I}} s_i$ is at most S and the total weight $\sum_{i \in \mathcal{I}} w_i$ is at most W.

LEMMA 5.3. *A simple greedy algorithm yields a constant factor approximation for the two-dimensional knapsack problem. Using dynamic programming we can obtain a $1 + \delta$-approximation for any fixed value $\delta > 0$.*

Proof. First notice that it is no loss of generality to assume that the size S and the weight limit W are both 1.

Consider items that have both size and weight at most $\frac{1}{2}$. We claim that the following greedy method provides an approximation: Find a greedy solution to maximizing using the sum $s_i + w_i$ as size, i.e., this greedy algorithm approximates the maximum possible value subject to the limit that the total size *plus* the total weight is at most 1. The optimal solution has size at most 1 and weight at most 1, so the sum of the total size and weight is at most 2. Hence, by Fact 5.2, the value obtained is at least $\frac{1}{4}$th of the optimal.

For items that have either size above $\frac{1}{2}$ or weight above $\frac{1}{2}$ at most 2 can fit in a knapsack, so we can get the optimal solution by trying all pairs.

The better of the two solutions obtained has a value of at least $\frac{1}{5}$th of the optimum. ∎

Next we consider large items. The definition of effective size is also related to $1 - g$: The effective size of an item of type (hi, lo, q) is defined as $\bar{s}$, where $\frac{1}{\bar{s}} = \min\{j : \ j(hi - lo) > 1 - g\}$, the number of copies the probabilistic part of this item can fit in a knapsack of size $1 - g$. Given an estimate for the value g, we group large items according to their effective size and pack items of one effective size using the two-dimensional knapsack problem above. As before, we separate items of identical effective size, depending on whether they have large or normal probability. For large probability items of effective size s, we need at most $\frac{1}{s} - 1$ items of total lo value at most g and maximum total value. For normal probability items one dimension is the lo value, where the total lo value is limited to g, and the other dimension is the probability, bounded by p^s/se. In both cases we can use Lemma 5.3 to obtain a knapsack solution.

Next we consider the issue of how many different estimates we have to consider in order to get a near-optimal solution using large items. Depending on the value of g the effective size of an item can change. Each item can have at most ε^{-1} different effective sizes, and hence it creates at most ε^{-1} different "cut-off" values for g. Hence, the grouping of large items changes for at most $n\varepsilon^{-1}$ discrete values of g, where n

is the number of large items. This implies that it suffices to try $O(n\varepsilon^{-1})$ different g values in order to get an approximately optimal solution.

The above discussion proves the following theorem.

THEOREM 5.4. *Let $X_1, \ldots, X_n$ be independent trials of the type defined above.*

- *There is a polynomial time algorithm that finds a solution to the knapsack problem with items $X_1, \ldots, X_n$ of value at least an $O(\log p^{-1})$ fraction of the optimum.*
- *For any $\varepsilon > 0$, there is a polynomial time algorithm that finds a solution to this knapsack problem using a knapsack size $1+\varepsilon$ and overflow probability at most p of value at least an $O(\varepsilon^{-1})$ fraction of the maximum possible with a knapsack of size 1 and overflow probability p.*
- *For any $\varepsilon > 0$, there is a polynomial time algorithm that finds a solution to this knapsack problem using a knapsack size 1 and overflow probability p of value at least an $O(\varepsilon^{-1})$ fraction of the maximum possible with a knapsack of size 1 and overflow probability $p^{1+\varepsilon}$.*

Bin-packing with other distributions. Using the knapsack result and set-cover we get a bin-packing algorithm for independent items of type (lo, hi, q). To get a solution to the bin-packing problem we repeatedly take the maximum number of items possible to include in a single bin. The result is an $O(\log n)$ extra factor in the approximation ratio.

COROLLARY 5.5. *Let $X_1, \ldots, X_n$ be independent trials of the type defined above.*

- *There is a polynomial time algorithm that finds a solution to the bin-packing problem with items $X_1, \ldots, X_n$ using at most $O(\log p^{-1} \log n)$ times the minimum possible number of bins.*
- *For any $\varepsilon > 0$, there is a polynomial time algorithm that finds a solution to this bin-packing problem using bins of size $1+\varepsilon$ and overflow probability at most p with a number of bins that is at most $O(\varepsilon^{-1} \log n)$ times the minimum possible with bins of size 1 and overflow probability p.*
- *For any $\varepsilon > 0$, there is a polynomial time algorithm that finds a solution to this bin-packing problem using bins size 1 and overflow probability p with at most $O(\varepsilon^{-1} \log n)$ times as many bins as the minimum possible with bins of size 1 and overflow probability $p^{1+\varepsilon}$.*

6. Extensions to general networks. The model we have been considering—two nodes communicating over a set of parallel links—is a common one in the study of bursty traffic. However, it is interesting to consider the extent to which one can carry over the results developed here to the problem of routing bursty connections in a general network. The model for a general network follows directly from the discussion in the introduction: we are given a graph $G = (V, E)$ with capacities $\{c_e\}$ on the edges, and source-sink pairs (s_i, t_i) indicating connection requests in the network. For each source-sink pair, we are given a random variable X_i corresponding to the demand of that connection; a *routing* is a choice of a path P_i in G for each connection (s_i, t_i).

There are several options for how one might want to model the *capacity constraints* for a problem of this type; we define two main possibilities here. Suppose we are given an allowed overflow probability p.

(i) The *link-based overflow constraint* requires that for each edge e, we have $\Pr[\sum_{i:e\in P_i} X_i > c_e] \le p$.

(ii) The *connection-based overflow constraint* requires that for each connection

(s_i, t_i), we have

$$\Pr\left[\exists e \in P_i : \sum_{i:e\in P_i} X_i > c_e\right] \leq p.$$

One can argue that from the perspective of providing guaranteed quality of service to users in a network, the *connection-based overflow constraint* is more natural. In this section we use this model.

Now suppose we are in a "high-capacity" setting in which the capacity of every edge exceeds the *peak* bandwidth rate of every connection X_i by a factor of $c\log(p^{-1}|E|)$ for an appropriate constant c. Let us define the *value* of a set of connections to be the sum of their expectations; we consider the problem of accepting a set of connections of maximum value. We run the on-line algorithm of Awerbuch, Azar, and Plotkin [2], using $\mathrm{E}[X_i]$ as the demand for connection (s_i, t_i) and $\frac{1}{4}c_e$ as the capacity of edge e. The analysis of [2] can then be used to show the following.

LEMMA 6.1. *For any constant γ there is a constant C such that, if k denotes the total value of connections accepted by the algorithm, then in any routing of a set of connections of value at least $C(\log|E|)k$, there is some edge e carrying a total expected value greater than γc_e.*

THEOREM 6.2. *The set of connections accepted by the above algorithm satisfies the connection-based overflow constraints, and the total value of the connections accepted is within an $O(\log|E|)$ factor of the off-line optimum on the graph G.*

Proof. Without loss of generality, we may assume that the minimum edge capacity in the network is 1. Recall our assumption that the peak rate of any connection X is at most $1/(c\log(p^{-1}|E|))$; thus, for each connection X, the effective bandwidth $\beta_{p|E|^{-1}}(X)$, with respect to probability $\frac{p}{|E|}$, is at most $2\mathrm{E}[X]$. Now Proposition 3.6 implies that our routing satisfies the link-based overflow constraint with probability $\frac{p}{|E|}$ and hence the connection-based overflow constraints with probability p.

To compare our performance to that of the optimum, we use Lemma 6.1 with $\gamma = 8$. Further, we give up a constant factor in the approximation ratio and use Lemma 3.10 to model each connection. Using the notation of the lemma we model a connection X as a sum of of independent Bernoulli trials $\frac{1}{2}Y = \sum_i \frac{1}{2}Y_i$ whose peak rates are inverse powers of two, such that $Y \leq X$ and $\mathrm{E}(Y) \leq \frac{1}{2}\mathrm{E}(X)$.

Lemma 4.5 shows that such a routing violates the link-based overflow constraint on edge e, and hence any path through the edge e violates the connection-based overflow constraint. It follows that our routing is within $O(\log|E|)$ of optimal. ∎

We note that the analysis of [2] also allows us to provide performance guarantees in terms of more general notions of "value."

REFERENCES

[1] J. ASPNES, Y. AZAR, A. FIAT, S. PLOTKIN, AND O. WAARTS, *On-line load balancing with applications to machine scheduling and virtual circuit routing*, in Proceedings of the 25th ACM Symposium on the Theory of Computing, San Diego, CA, 1993, pp. 623–631.

[2] B. AWERBUCH, Y. AZAR, AND S. PLOTKIN, *Throughput-competitive online routing*, in Proceedings of the 34th IEEE Foundations in Computer Science, Palo Alto, CA, 1993, pp. 32–40.

[3] Y. BARTAL, A. FIAT, H. KARLOFF, AND R. VORHA, *New algorithms for an ancient scheduling problem*, in Proceedings of the 24th ACM Symposium on the Theory of Computing, Vancouver, British Columbia, Canada, 1992, pp. 51–58.

[4] A. ELWALID AND D. MITRA, *Effective bandwidth of general Markovian traffic sources and admission control of high speed networks*, IEEE Trans. Networking, 1 (1993), pp. 329–343.

[5] R. Gawlick, C. Kalmanek, and K. G. Ramakrishnan, *On–line routing for permanent virtual circuits*, in Proceedings of INFOCOM, Boston, MA, 1995, pp. 278–288.

[6] R. Gibbens and P. Hunt, *Effective bandwidths for the multi-type UAS channel*, Queueing Systems Theory Appl., 9 (1991), pp. 17–27.

[7] R. L. Graham, *Bounds for certain multiprocessing anomalies*, Bell System Tech. J., 45 (1966), pp. 563–1581.

[8] R. L. Graham, *Bounds on multiprocessing anomalies*, SIAM J. Appl. Math., 17 (1969), pp. 416–429.

[9] D. S. Hochbaum and D. B. Shmoys, *Using dual approximation algorithms for scheduling problems: Theoretical and practical results*, J. ACM, 34 (1987), pp. 144–162.

[10] J. Y. Hui, *Resource allocation for broadband networks*, IEEE J. Selected Areas in Comm., 6 (1988), pp. 1598–1608.

[11] F. P. Kelly, *Effective bandwidths at multi-class queues*, Queueing Systems Theory Appl., 9 (1991), pp. 5–15.

[12] F. P. Kelly, *Notes on effective bandwidths*, in F. P. Kelly, S. Zachary, I. B. Zeidins, eds., Stochastic Networks: Theory and Applications, Oxford University Press, Oxford, 1996, pp. 141–168.

[13] F. P. Kelly, S. Zachary, and I. B. Zeidins, eds., *Stochastic Networks: Theory and Applications*, Oxford University Press, Oxford, 1996.

[14] T. Lehtonen, *Scheduling jobs with exponential processing times on parallel machines.* J. Appl. Probab., 25 (1988), pp. 752–762.

[15] M. Pinedo, *Scheduling: Theory, Algorithms and Systems*, Prentice-Hall, Upper Saddle River, NJ, 1995.

[16] S. Plotkin, *Competitive routing in ATM networks*, IEEE J. Selected Areas in Communications, 13 (1995), pp. 1128–1136.

[17] P. Raghavan and C. D. Thompson, *Randomized rounding: A technique for provably good algorithms and algorithmic proofs*, Combinatorica, 7 (1987), pp. 365–374.

[18] G. de Veciana and J. Walrand, *Effective bandwidths: Call admission, traffic policing and filtering for ATM networks*, Queueing Systems Theory Appl., 20 (1995), pp. 37–39.

[19] G. Weiss, *Approximation results in parallel machines stochastic scheduling*, Ann. of Oper. Res., 26 (1990), pp. 195–242.

[20] G. Weiss, *A tutorial in stochastic scheduling*, in Scheduling Theory and Its Applications, P. Chretienne, E. G. Coffman, J. K. Lenstra, and Z. Liu, eds., Wiley, New York, 1995, pp. 33–64.