# Stochastic Load Balancing and Related Problems

ASHISH GOEL *
University of Southern California

PIOTR INDYK †
Stanford University

## Abstract

*We study the problems of makespan minimization (load balancing), knapsack, and bin packing when the jobs have stochastic processing requirements or sizes. If the jobs are all Poisson, we present a two approximation for the first problem using Graham's rule, and observe that polynomial time approximation schemes can be obtained for the last two problems. If the jobs are all exponential, we present polynomial time approximation schemes for all three problems. We also obtain quasi-polynomial time approximation schemes for the last two problems if the jobs are Bernoulli variables.*

## 1 Introduction

In traditional scheduling problems, each job has a known deterministic size and duration. There are cases, however, where the exact size of a job is not known at the time when a scheduling decision needs to be made; all that is known is a probability distribution on the size of the job. Given a schedule, the value of the objective function itself becomes a random variable. The goal then is to find a schedule such that the expected value of the objective function is minimized or maximized, as the case may be.

Kleinberg, Rabani, and Tardos [7] studied the following three problems for a restricted class of job length distributions.

**Stochastic Load Balancing:** Consider $n$ jobs which need to be allocated on $m$ identical machines. The size of the $i$-th job is $X_i$, where the $X_i$s are independent random variables. Let $M_i$ refer to the machine to which the $i$-th job gets allocated. The stochastic load balancing problem is the problem of allocating jobs to machines such that the expected value of the maximum load on a machine is minimized, i.e. the quantity

$$L = \mathbf{E}\left[\max_{j=1}^{m} \sum_{i:M_i=j} X_i\right]$$

gets minimized.

**Stochastic Knapsack:** The stochastic knapsack problem is the following: Given a set $S$ of $n$ independent random variables $\{X_1, \ldots, X_n\}$ with associated profits $\{v_1, \ldots, v_n\}$, and an *overflow probability* $p$, pick a subset $P$ of $S$ such that the objective function

$$Z = \sum_{i:X_i \in P} v_i$$

is maximized under the constraint

$$\mathbf{Pr}\left[\sum_{i:X_i \in P} X_i > 1\right] \leq p.$$

We use $Z^*$ to refer to this optimum value.

**Stochastic Bin Packing:** The stochastic bin packing problem is the following: given a set of independent random variables $S = \{X_1, X_2, \ldots, X_n\}$ and an overflow probability $p$, partition the set $S$ into the smallest number of sets $S_1, \ldots, S_k$ such that $\mathbf{Pr}\left[\sum_{i:X_i \in S_j} X_i > 1\right] \leq p$ for all $1 \leq j \leq k$. This is a natural analog of the deterministic bin packing problem.

These problems have potential applications wherever the deterministic counterparts are used. We would like to distinguish these problems from those solved by traditional probabilistic algorithms. Probabilistic algorithms take their input from a probability distribution; these problems take a probability distribution as input. Our work, following on the lines of Kleinberg *et al.* [7], studies these problems in an approximation algorithms setting.

### 1.1 Previous work

Kleinberg *et al.* [7] obtained an $O(1)$ approximation algorithm for the load balancing problems where the random

*Department of Computer Science, University of Southern California. The author was at Stanford University when this research was conducted.
Research supported by ARO Grants DAAG55-98-1-0170 and ASSERT award DAAG55-97-1-0221.
Email: agoel@cs.usc.edu

†Department of Computer Science, Stanford University. Email: indyk@cs.stanford.edu

variables $X_i$ have an arbitrary distribution[1]. Kleinberg *et al*. focus primarily on obtaining an $O(1)$-approxmation algorithm and do not attempt to optimize the constant of approximation. A summary of our results is presented in Table 1. Weiss, and Weiss and Pinedo looked at the case where jobs arrive in an online fashion and are distributed according to Poisson or exponential distributions [13, 15, 14, 11]. Lehtonen looked at the special case of scheduling identical exponential jobs on two non-homogeneous machines [9]. Weber also studied the problem of scheduling jobs with stochastic processing time on parallel machines [12] in a continuous model i.e. new jobs are scheduled when old ones finish.

### 1.2 Our Results

We study the cases where the job sizes are all Poisson, all exponential, or all Bernoulli. For Poisson jobs, we obtain a Polynomial Time Approximation Scheme (PTAS) for the knapsack and bin packing problems by a simple reduction to the deterministic case. We then prove that Graham's rule works for Poisson jobs, obtaining a 2 approximation for the load balancing problem. For exponential jobs, we obtain a PTAS for each of the three problems using dynamic programming along the lines of [6, 8, 5]. For Bernoulli jobs our results are weaker. We obtain Quasi-Polynomial Time Approximation Schemes (QPTAS) for the bin packing and knapsack[2] variants; the QPTAS also assumes that $p$ is a constant. Table 1 summarizes our results. For the knapsack and bin packing problems, both our algorithms and those in [7] relax the constraints by a factor of $1 + \epsilon$. The algorithms presented in [7] also work in an online setting. Our algorithms for Poisson distributions work in an online setting, but those for exponential and Bernoulli distributions don't.

The case where the job sizes are exponential is particularly interesting. This corresponds to allocating jobs to different (but identical) servers, where the holding time of each job is exponential (albeit with different means). This is a popular queueing system. Thus our results for the exponential case have applications to minimizing the finishing time for multi-server queueing systems, where jobs must be distributed to the individual server queues in advance.

## 2 Graham's Rule for Poisson Jobs

We first look at the Stochastic Knapsack and Bin Packing problems where the size of the $i$-th job is Poisson with mean $\mu_i$. Let $\mathcal{P}(\mu)$ denote the Poisson variable with mean $\mu$. We will use the following elementary fact repeatedly:

$$\mathcal{P}(x_1) + \mathcal{P}(x_2) \equiv_D \mathcal{P}(x_1 + x_2),$$

where $X \equiv_D Y$ indicates that the random variables $X$ and $Y$ have identical distributions. Hence, the random variable corresponding to the sum of jobs in a bin/knapsack is just a Poisson variable with mean equal to the sum of means of the jobs in the bin/knapsack. Find $\mu_p$ such that $\mathbf{Pr}\left[\mathcal{P}(\mu_p) > 1\right] = p$. The constraint that a bin/knapsack not overflow with probability more than $p$ now translates to the constraint that the sum of the means of the random variables packed in the bin/knapsack not be larger than $\mu_p$. This gives rise to a simple approximation scheme which uses the PTAS for the deterministic version of knapsack/bin-packing problems [6, 8, 5], with the size of each bin/knapsack set to $\mu_p$ and each Poisson random variable replaced by a deterministic variable with the same mean; the details are omitted from this version. Also, we can use the corresponding deterministic online algorithms [2] to obtain online algorithms for the case where the input distributions are Poisson[3].

The above simple solutions suggest that the techniques used for deterministic load balancing may also apply to Stochastic Load Balancing with Poisson variables. This is indeed the case.

**Graham's rule for load balancing with deterministic variables [4]** *Arrange all jobs in an arbitrary order. Pick these jobs one by one and allocate them to the bin which has currently the smallest load.*

Graham's rule results in a $2 - 1/m$ approximation for deterministic jobs. The natural equivalent of Graham's rule for Poisson variables would be to replace each variable by a deterministic variable with the same mean, and then run Graham's algorithm.

**Theorem 2.1** *Graham's rule results in a* $2$ *approximation for the Stochastic Load Balancing problem with Poisson variables.*

The proof follows from a series of lemmas. We first present the main technical lemma. If $X$ and $Y$ are two random variables, $X$ is said to stochastically dominate $Y$ (denoted $Y \leq_{\text{sd}} X$) if for each $x$, $\mathbf{Pr}\left[Y \leq x\right] \geq \mathbf{Pr}\left[X \leq x\right]$.

**Lemma 2.1** *Let* $x_1 > x_2$ *and* $2\delta \leq x_1 - x_2$. *Let* $M_1 = \max(\mathcal{P}(x_1 - \delta), \mathcal{P}(x_2 + \delta))$ *and* $M_2 = \max(\mathcal{P}(x_1), \mathcal{P}(x_2))$. *Then,* $M_1 \leq_{sd} M_2$.

**Proof:** Let $M(x, y)$ represent $\max(\mathcal{P}(x), \mathcal{P}(y))$. Let $f_t(x)$ represent $\mathbf{Pr}\left[\mathcal{P}(x) \leq t\right]$. Then $\mathbf{Pr}\left[M(x, y) \leq t\right] = f_t(x) f_t(y) = e^{\log f_t(x) + \log f_t(y)}$. We now show that

[1]A cursory reading of [7] might seem to indicate that their algorithm only works under the assumption that for all random variables $X_i$, the quantity $\mathbf{E}\left[m^{X_i}\right]$ is defined. However their algorithms work for arbitrary distributions.

[2]A PTAS for Stochastic Knapsack with Bernoulli jobs can be obtained using techniques recently developed by Chekuri *et al*. [3].

[3]For the knapsack problem this requires additional restrictions on job sizes and profits

| Problem | Poisson Job Sizes | Exponential Job Sizes | Bernoulli Job Sizes |
|---|---|---|---|
| Load Balancing | 2-approx | PTAS | NA |
| Knapsack | PTAS | PTAS | QPTAS |
| Bin Packing | PTAS | PTAS | QPTAS |

**Table 1. The approximation ratios obtained in this paper.**

$\log f_t(x)$ is concave with respect to $x$. Recall that $f_t(x) = e^{-x}\sum_{k=0}^{t} \frac{x^k}{k!}$. A simple computation shows that

$$\frac{\mathbf{d}\log f_t(x)}{\mathbf{d}x} = \frac{\frac{\mathbf{d}f_t(x)}{\mathbf{d}x}}{f_t(x)} = -\frac{x^t/t!}{\sum_{k=0}^{t}(x^k/k!)}$$

But $\frac{x^t/t!}{\sum_{k=0}^{t}(x^k/k!)}$ is the Erlang-B formula [1] for blocking on a link with capacity $t$ and arrival rate $x$; this is known to be monotonically increasing in $x$. Hence $\frac{\mathbf{d}\log f_t(x)}{\mathbf{d}x}$ is monotonically decreasing i.e. $\log f_t(x)$ is concave with respect to $x$. This implies that if $x + y$ is fixed, then $\log f_t(x) + \log f_t(y)$ increases as $|y - x|$ decreases. Therefore, $\mathbf{Pr}[M_1 \le t] \ge \mathbf{Pr}[M_2 \le t]$ for all $t$ which, by definition, implies $M_1 \le_{\text{sd}} M_2$. $\square$

Let $S_j$ be the sum of the means of the random variables allocated to bin $j$. Let $x_j$ be the last variable allocated to bin $j$. Let $y_j = S_j - x_j$. Let $M^*$ represent the optimum value of the objective function. Define $\overline{\mu} = \frac{\sum_i \mu_i}{m}$. From the definition of Graham's rule, it follows that $y_j \le \overline{\mu}$.

**Lemma 2.2** $\mathbf{E}[\max(\mathcal{P}(y_1), \ldots, \mathcal{P}(y_m))] \le M^*$

**Proof:** From Lemma 2.1 it follows that $\mathbf{E}[\max(\mathcal{P}(\overline{\mu}), \mathcal{P}(\overline{\mu}), \ldots \text{repeated } m \text{ times})] \le M^*$; the details are omitted. Further, $y_j \le \overline{\mu} \Rightarrow \mathcal{P}(y_j) \le_{\text{sd}} \mathcal{P}(\overline{\mu})$, or $\mathbf{E}[\max(\mathcal{P}(y_1), \ldots, \mathcal{P}(y_m))] \le \mathbf{E}[\max(\mathcal{P}(\overline{\mu}), \mathcal{P}(\overline{\mu}), \ldots \text{repeated } m \text{ times})] \le M^*$. $\square$

The following lemma is obvious (for arbitrary distributions, not just Poisson).

**Lemma 2.3** $\mathbf{E}[\max(\mathcal{P}(x_1), \ldots, \mathcal{P}(x_m))] \le M^*$

Now, we prove that Graham's rule gives a 2 approximation.

$$\begin{aligned}
&\mathbf{E}[\max(\mathcal{P}(S_1), \ldots \mathcal{P}(S_m))] \\
&= \mathbf{E}[\max(\mathcal{P}(x_1 + y_1), \ldots \mathcal{P}(x_m + y_m))] \\
&\le \mathbf{E}[\max(\mathcal{P}(x_1) \ldots \mathcal{P}(x_m))] + \mathbf{E}[\max(\mathcal{P}(y_1) \ldots \mathcal{P}(y_m))] \\
&\le 2M^*
\end{aligned}$$

## 3 Stochastic Load Balancing with Exponential Jobs

In this section we consider the Stochastic Load Balancing problem under the assumption that the $i$-th job, $X_i$, is an exponential random variable with mean $\lambda_i$ ($\mathbf{Pr}[X_i < x] = 1 - e^{-x/\lambda_i}$). We present a Polynomial Time Approximation Scheme (PTAS) for this case.

We first assume that the optimum allocation results in $L \le 1$. Let $\mathcal{S}$ denote the space consisting of all subsets of $\{X_1, \ldots, X_n\}$. Let $\mathcal{D}$ denote the space consisting of all possible distributions for the value of the maximum load in any allocation of any set $S \in \mathcal{S}$ of variables to any number $1 \le k \le m$ of machines. Define the function $f : \mathcal{S} \times \mathcal{D} \times \{1, \ldots, m\} \to \{0, 1\}$ as follows.

$$\begin{aligned}
f(S, D, k) &= 1 \quad \textit{if the set } S \textit{ of variables can be allocated to } k \textit{ machines resulting in a distribution } D \textit{ for the maximum load on a machine,} \\
&= 0 \quad \textit{otherwise.}
\end{aligned}$$

The function $f$ can be computed using dynamic programming as follows. Suppose $f(\,\cdot\,, \,\cdot\,, k)$ has been calculated. Initialize $f(\,\cdot\,, \,\cdot\,, k+1)$ to zero. For all $S, S' \in \mathcal{S}, D' \in \mathcal{D}$ such that $S$ and $S'$ are disjoint, and $f(S', D', 1) = 1$,

1. Compute $D_S$, the distribution of the sum of random variables in the set $S$.
2. Compute the distribution $D = \max(D, D_S)$.
3. Set $f(S \cup S', D, k+1) = 1$.

The running time of the above algorithm is $O(|\mathcal{S}|^2 \cdot |\mathcal{D}| \cdot m \cdot (T_1 + T_2))$ where $T_1$ and $T_2$ are the times needed to implement steps 1 and 2 above. $|\mathcal{S}| = 2^n$, and $|\mathcal{D}|$ is even larger. Therefore the above algorithm is quite inefficient. The rest of this section is devoted to demonstrating that given a fixed constant $\epsilon > 0$, both the variable space $\mathcal{S}$ and the distribution space $\mathcal{D}$ (section 3.2) can be approximated and discretized so that their sizes become polynomial in $n$ and $m$ (the sizes are exponential in $1/\epsilon$). Section 3.1 obtains a polynomial sized variable space and proves that this approximation results in a multiplicative error of at most $1 + \epsilon$ in the value of the objective function; Section 3.2 does the same for the distribution space and also shows that $T_1$ and $T_2$ are polynomial in $n$ and $m$. We now have to remove the assumption that $L \le 1$. We know that $\frac{1}{m}\sum_i \lambda_i \le L \le \sum_i \lambda_i$. Thus we execute the above scheme for all candidate values of $L$ of the form $(\frac{1}{m}\sum_i \lambda_i) \cdot (1+\epsilon)^j$ for $j$ ranging from $0$ to $\lceil \log_{1+\epsilon} m \rceil$, rescaling the problem (i.e. $\lambda_i$s) each time so that $L = 1$. For small $\epsilon$, the number of such values is $O(\frac{1}{\epsilon}\log m)$. Of all the objective function values obtained during this procedure, we pick the best. This completes the description of the PTAS.

### 3.1 Obtaining polynomial variable space

The basic idea is to first discretize jobs so that they have means that differ by powers of $1+\epsilon$. Then the jobs are divided into small and big; the small jobs are treated as carrying deterministic load; the size of the variable space restricted to big jobs is shown to be small.

Clearly $\lambda_i \le 1$ for all jobs $i$. A *canonical* random variable of type $k$ is an exponential variable with mean of the form $(1+\epsilon)^{-k}$, where $k \ge 0$. For each big job $i$, decrease $\lambda_i$ by the minimum amount needed to make $X_i$ a canonical variable. Let $n_k$ be the number of canonical variables of type $k$.

**Lemma 3.1** *If* $\gamma = \dfrac{e}{e-1}$*, then* $n_k \le e^{\gamma(1+\epsilon)^k}$.

**Proof:** Let $Y_1, Y_2, \ldots, Y_{n_k}$ represent the $n_k$ canonical variables of type $k$. Define $P = \mathbf{Pr}\left[\max(Y_1,\ldots,Y_{n_k}) \ge \gamma\right]$. The quantity $P\cdot\gamma$ is a lower bound on the quantity $L$, and since $L \le 1$, $P \le 1/\gamma$. Let $p = \mathbf{Pr}\left[Y_1 \ge \gamma\right]$ and $\alpha = n_k \cdot p$. Since the $Y_j$ are i.i.d., $P = 1-(1-p)^{n_k} = 1-(1-p)^{\alpha/p} \ge 1-e^{-\alpha}$. Therefore

$$\begin{array}{ll} & \gamma(1-e^{-\alpha}) \le 1 \\ \text{i.e.} & e^{-\alpha} \ge 1 - 1/\gamma \\ \text{i.e.} & e^{-\alpha} \ge e^{-1} \\ \text{i.e.} & \alpha \le 1 \\ \text{i.e.} & n \le 1/p \end{array}$$

Since $p = e^{-\gamma/(1+\epsilon)^{-k}}$, $n \le e^{\gamma\cdot(1+\epsilon)^k}$ □

The $i$-th job is called *small* if $\lambda_i < \epsilon^3/\log(n/\epsilon)$, and *big* otherwise. There are at most $\beta = O(\log_{1+\epsilon}(\log n)/\epsilon)$ types of big canonical variables. Let $S$ be a subset of $X_1,\ldots,X_n$. Let $\lambda_S = \sum_{X_i \in S, X_i \text{ is small}} \lambda_i$. Let $n_{k,S}$ denote the number of canonical variables of type $k$ in $S$. A canonical representation of $S$ is $\left(\left\lceil \dfrac{\lambda_S}{\epsilon}\right\rceil, n_{0,S}, n_{1,S}, \ldots, n_{\beta-1,S}\right)$.

**Lemma 3.2** *The number of possible canonical representations is* $(n/\epsilon)^{O(1/\epsilon^5)}$.

**Proof:** Since each small job has mean at most $\epsilon/\log n$, and there are at most $n$ small jobs, $\lambda_S \le n\epsilon/\log n$, which implies that the number of possible different values of $\lceil \frac{\lambda_S}{\epsilon}\rceil$ is at most $\lceil n/\log n\rceil < n$. Since $0 \le n_{k,S} \le n_k$, the total number of canonical representations is at most

$$\begin{aligned} & n\prod_{k=0}^{\beta-1}(1+n_k) \\ \le\ & n\cdot 2^{\beta}\prod_{k=0}^{\beta-1} n_k \\ \le\ & n\cdot 2^{\beta} e^{\gamma\sum_{k=0}^{\beta-1}(1+\epsilon)^k} \\ \le\ & n\cdot 2\cdot(\log n/\epsilon^4)^{\frac{1}{\log(1+\epsilon)}}\cdot e^{\gamma(1+\epsilon)^{\beta}/\epsilon} \\ =\ & (n/\epsilon)^{O(1/\epsilon^5)}. \end{aligned}$$

□

We are going to restrict $\mathcal{S}$ to contain canonical representations of variables. The first component $\lambda_S$ will be interpreted as a deterministic variable of mean $\lambda_S$.

**Theorem 3.1** *The above approximation to the variable space can increase the objective function value by at most* $1+c\epsilon$ *for some constant c.*

**Proof:** For each job, the mean increases by no more than $(1+\epsilon)$. Thus the sum of all the job sizes on a machine scale up by at most $1+\epsilon)$; hence the maximum of the total load on each machine also scales up by at most $1+\epsilon$. The only other change is due to approximating small variables by their sum, $\lambda_S$. This change can be bounded by $1+\epsilon$ by a simple application of the Chernoff Bound [10]; the details are omitted from this version. □

### 3.2 Obtaining polynomial distribution space

We now define the concept of a *truncated distribution*. Let $X_1 \ldots X_t$ denote the random variables defining the sizes of jobs allocated to a fixed machine; let $X = X_1 + \ldots X_t$.

We define $\overline{X}$ to be a random variable taking values from the set $\{\epsilon, \epsilon(1+\epsilon), \epsilon(1+\epsilon)^2 \ldots C\}$, for a constant $C = C(\epsilon) > 1$ defined later. The distribution of $\overline{X}$ is defined as

$$\mathbf{Pr}\left[\overline{X} = \epsilon(1+\epsilon)^i\right] = \mathbf{Pr}\left[X \in [\epsilon(1+\epsilon)^i, \epsilon(1+\epsilon)^{i+1})\right]$$

for $i \ge 0$; we define $\mathbf{Pr}\left[\overline{X} = C\right] = \mathbf{Pr}\left[X \ge C\right]$. We call $\overline{X}$ a truncated version of $X$. In the following we show that replacing $X$ by $\overline{X}$ does not change significantly the value of the objective function, thus we can use truncated distribution for the purpose of solving our problem. The advantage of using truncated distribution is that (after discretizing the probabilities with step $D = 1/\text{poly}(n)$) the number of distinct truncated distribution is roughly $n^{O(\log_{1+\epsilon}(C/\epsilon))}$, i.e. polynomial. Precision up to $D$ is sufficient for our purpose; details are omitted from this version.

The key property which we exploit is formulated in the following lemma.

**Lemma 3.3** *Let $X^1 \dots X^m$ be the random variables denoting the sizes of jobs allocated to machines $1 \dots m$. Then for $C = O(\log 1/\epsilon)$*

$$E[\max_i \overline{X}^i] \le E[\max_i X^i] \le (1 + O(\epsilon))E[\max_i \overline{X}^i].$$

The first part of the inequality is easy to see, thus we will focus on the second one. The proof will proceed in a sequence of lemmas.

**Lemma 3.4** *Let $x > 0$ be such that $\sum_i \mathbf{Pr}\left[X^i > x\right] = 1$. Then $E[\max_i X^i] \ge x(1 - 1/e)$.*

**Proof:** By observing that

$$\begin{aligned} \mathbf{Pr}\left[\max_i X^i \le t\right] &= \prod_i \mathbf{Pr}\left[X^i \le t\right] \\ &= \prod_i (1 - \mathbf{Pr}\left[X^i > t\right]) \\ &\ge 1/e \end{aligned}$$

□

**Lemma 3.5** *For any $i$ and $\alpha, \beta > 0$ such that $\beta \ge 3E[X^i]$ and $\alpha \ge \beta$ we have*

$$\frac{\mathbf{Pr}\left[X^i \ge \alpha\right]}{\mathbf{Pr}\left[X^i \in [\alpha - \beta, \alpha]\right]} \le 1/2.$$

**Proof:** The key property of $X^i$ which we use here is that

$$\mathbf{Pr}\left[X^i \ge \alpha | X^i \ge \alpha - \beta\right] \le \mathbf{Pr}\left[X^i \ge \beta\right]$$

which follows from the fact that $X^i$ is a sum of variables with memoryless distribution. By Markov inequality, the RHS can be upper bounded by $1/3$. The inequality can be then rewritten as

$$\frac{\mathbf{Pr}\left[X^i \ge \alpha\right]}{\mathbf{Pr}\left[X^i \ge \alpha\right] + \mathbf{Pr}\left[\alpha \ge X^i \ge \alpha - \beta\right]} \le 1/3$$

which implies

$$2\mathbf{Pr}\left[X^i \ge \alpha\right] \le \mathbf{Pr}\left[\alpha \ge X^i \ge \alpha - \beta\right]$$

which was to be shown. □

**Lemma 3.6** *Let $y = E[\max_i X^i]/(1 - 1/e)$. Then for any $j > 0$*

$$\sum_i \mathbf{Pr}\left[X^i \in [jy, (j+3)y]\right] \le 1/2^{\lceil j/3 - 1 \rceil}.$$

**Proof:** Observe that

$$\sum_i \mathbf{Pr}\left[X^i \in [jy, (j+3)y]\right] \le \sum_i \mathbf{Pr}\left[X^i \in [y, 3y]\right] \cdot \prod_{k=1}^{\lceil j/3-1 \rceil} \frac{\mathbf{Pr}\left[X^i \ge (3k+1)y\right]}{\mathbf{Pr}\left[X^i \in [(3k-2)y, (3k+1)y]\right]}$$

Since $y \ge E[\max_i X^i] \ge E[X^i]$ we get (by Lemma 3.5)

$$\frac{\mathbf{Pr}\left[X^i \ge (3k+1)y\right]}{\mathbf{Pr}\left[X^i \in [(3k-2)y, (3k+1)y]\right]} \le 1/2$$

and since (by Lemma 3.4) $y \ge x$ we have

$$\sum_i \mathbf{Pr}\left[X^i \in [y, 3y]\right] \le 1$$

we are done. □

The proof of Lemma 3.3 now follows easily from Lemma 3.6, since the difference $|E[\max_i \overline{X}^i] - E[\max_i X^i]|$ is bounded by

$$O(\epsilon + \sum_{k \ge C} \sum_i \mathbf{Pr}\left[\max_i X^i \in [ky, (k+1)y]\right](k+1))$$

which for $C = O(\log 1/\epsilon)$ is $O(\epsilon)$.

**The running times $T_1$ and $T_2$.** Recall that $T_1$ is the time needed to compute the distribution of the sum of random variables in the set $S$ and $T_2$ is the time of computing the distribution of $Z = \max(X, Y)$ where $X$ and $Y$ are random variables with the same support set $S$. Since we represent all probabilities with accuracy $\pm 1/\text{poly}(n)$, each probability fits within $O(\log n)$ bits and therefore we can assume that each operation on such numbers are performed in unit time. In this case it is easy to verify that computing the distribution of $Z$ takes $O(S)$ time, which gives a bound for $T_2$. For $T_1$, we discretize and approximate the distribution; a precision of $\epsilon/n$ in the value, and a precision of $1/\text{poly}(n)$ is sufficient in the probability. Thus an exponential variable can be converted into a discrete, approximate version in polynomial time. Finding the sum of at most $n$ such distributions can again be done in polynomial time (in fact each addition can be done in roughly linear time using Fast Fourier Transforms).

# 4 Stochastic Knapsack and Bin Packing

## 4.1 Stochastic Knapsack with Exponential Jobs

In this section we concentrate on the case where each $X_i$ is an exponential random variable; let $\lambda_i$ be the mean of $X_i$. For this case, we present an algorithm that, given a fixed constant $\epsilon < 1/2$, finds a subset $P$ of $S$ such that

1. $Z = \sum_{i:X_i \in P} v_i \geq Z^*$,

2. **Pr** $\left[\sum_{i:X_i \in P} X_i > 1+\epsilon\right] \leq p(1+\epsilon)$,

and runs in time polynomial in $n$ and $\log \frac{1}{p}$.

This scheme is a slightly different kind of PTAS; the constraints are violated by a factor of $1+\epsilon$ but the objective function value is no worse than the optimum. The basic idea is to divide the random variables into small and large; a variable is called small if it has mean less than $\epsilon^3/(\log 1/\epsilon \log 1/p$ and big otherwise. We treat the small and large variables differently. We assume that $p \leq 1/2$ and $\epsilon \leq 1/2$. The basic idea is to assume a knapsack of size $1+\epsilon$ and then, for all possible ways (with a tolerance of $\epsilon$) of splitting this capacity among the small and the large variables, compute separately the optimum allocation for the big and the small variables.

### Small variables

For $i = 0, 1, \ldots, \lceil n/\epsilon \rceil$, solve the deterministic fractional knapsack problem with a knapsack of size $\epsilon \cdot i$ and with just the small variables, pretending that the small variable $X_i$ is a deterministic variable with size $\lambda_i$. At most one variable gets fractionally chosen; choose this variable as well as all variables that get chosen as a whole.

**Lemma 4.1** *The total running time of this procedure is* $O(\frac{n}{\epsilon} \log n)$

**Proof:** To solve a fractional knapsack problem with capacity $\alpha$, it suffices to arrange the variables in decreasing order of $v_i/\lambda_i$ and then find the smallest $k$ such that the sum of the means for the first $k$ guys is larger than $x$. The sorting needs to be done once; $k$ can be found by doing a binary search. Therefore the total time is $O(n \log n) + O(\frac{n}{\epsilon} \log n)$. □

For any of the fractional knapsack solutions above, let $\mu$ be the sum of the means of the variables chosen, and let $X$ represent the random variable corresponding to the sum of these variables. Then, using Chernoff bounds, we obtain the following lemma.

**Lemma 4.2** **Pr** $[X > (1+\epsilon)\mu] \leq \epsilon p$

Intuitively, the above lemma shows that we do not violate the constraints by too much in the process of treating the variables as deterministic.

### Large variables

**Lemma 4.3** *Any variable with mean larger than* $1/\log(1/p)$ *can not be part of the solution.*

**Proof:** Suppose $\lambda_i > 1/\log(1/p)$. Then **Pr** $[X_i > 1] = e^{-1/\lambda_i} > e^{-\log(1/p)} = p$ i.e. $X_i$ alone violates the constraint. □

Henceforth, we will assume that $\lambda_i \leq 1/\log(1/p)$.

Round down the means of the remaining large variables to the largest value of the form $\frac{1/\log(1/p)}{(1+\epsilon)^i}, i \geq 0$. The number of distinct values for the means is at most

$$O\left(\log_{1+\epsilon} \frac{1/\log(1/p)}{\epsilon^3/(\log 1/\epsilon \log 1/p)}\right) = O\left(\frac{1}{\epsilon}\log(1/\epsilon)\right).$$

The number of variables which have the same mean and are part of the solution can be at most $\gamma = (1/\log 2)(\log 1/\epsilon \log 1/p/)\epsilon^3$ (The probability that the sum of any number of exponential variables is less than $1/\log 2$ times the sum of the means is at most $1/2$.) Therefore the number of candidate sets of big variables that are potential solutions is at most $\gamma^{O\left(\frac{1}{\epsilon}\log(1/\epsilon)\right)}$

The dynamic program is the following: for each value $x$ for which the small variable solution has been computed, consider a knapsack of capacity $1+\epsilon-x$. For this capacity, and for each big variable combination, compute the probability (up to an accuracy of $\epsilon p$) of the sum of these variables exceeding the knapsack capacity. If this probability is less than $p(1+\epsilon)$ then find the best set of variables corresponding to this combination. For example, if there are three variables of the same mean in the variable combination, then pick the three most profitable variables with this mean. Pick the combination with the maximum profit. Then, pick the $x$ which maximizes the combined profit of the small and big jobs.

Let $T$ be the time to compute the overflow probability, given a knapsack of size $k$.

**Lemma 4.4** $T = O\left(n^3/\epsilon \log \frac{n}{\epsilon p}\right)$

**Proof:** Represent each variable $X_i$ by a discrete random variable $\overline{X}_i$ which takes values of the form $(\epsilon/n) \cdot k$, where $k$ ranges from 0 to $\lceil n/\epsilon + 1 \rceil$ and

$$\begin{aligned} \mathbf{Pr}\left[\overline{X}_i = \epsilon/n) \cdot k\right] &= \mathbf{Pr}\left[(\epsilon/n) \cdot k \leq X_i < (\epsilon/n) \cdot (k+1)\right] \\ & \qquad \textit{if } k < \lceil n/\epsilon + 1 \rceil, \\ &= \mathbf{Pr}\left[(\epsilon/n) \cdot k < X_i\right] \qquad \textit{otherwise.} \end{aligned}$$

$O(n^2/\epsilon^2)$ multiplications[4] are needed to compute the sum of two such distributions (assuming that if the sum is larger than $1+\epsilon$, it will be set to $1+\epsilon$). To compute the sum of (at most $n$) such variables takes $O(n^3/\epsilon^2)$ multiplications, where each multiplication takes $O(\log \frac{n}{\epsilon p})$ time, since the final accuracy desired is $\epsilon p$. □

The proof of each of the following lemmas follows from the discussion above in a straightforward fashion and is omitted.

[4] Again, this can be done in nearly linear time using Fast Fourier Transforms.

**Lemma 4.5** *The running time of the algorithm is* $O(n^3 \log n) \cdot O\left(\frac{\log 1/\epsilon}{\epsilon^3} \log \frac{1}{\epsilon p}\right) \cdot \gamma^{O\left(\frac{\log 1/\epsilon}{\epsilon}\right)}$

While this is not quite a Fully Polynomial Time Approximation Scheme, it has the nice property that the terms involving $n$ are separate from terms involving $1/p$ and $1/\epsilon$. Also notice that the above algorithm is not strongly polynomial, since it has terms which are polynomial in $\log 1/p$, where $p$ is part of the input to the problem.

**Lemma 4.6** *There exist constants $c_1$ and $c_2$ such that the above algorithm finds a set $P$ of variables such that $\sum_{i:X_i \in P} v_i \geq Z^*$, and* $\mathbf{Pr}\left[\sum_{i:X_i \in P} X_i > 1 + c_1\epsilon\right] \leq p(1 + c_2\epsilon)$.

This completes the description of the PTAS.

## 4.2 Stochastic Knapsack with Bernoulli Jobs

In this section we concentrate on the case where each $X_i$ is a random variable with Bernoulli distributions, i.e. such that $\mathbf{Pr}[X_i = x_i] = p_i$ and $\mathbf{Pr}[X_i = 0] = 1 - p_i$, for some $x_i > 0$ and $0 \leq p_i \leq 1$. Since the techniques are similar to the ones used in the previous section, we only sketch the arguments. As before, we split the variables into small and large according to the value of $x_i$, with the threshold $t = \epsilon^3/(\log \frac{1}{\epsilon} \log \frac{1}{p})$ (this value, as in the case of exponential jobs, is chosen to allow application of Chernoff bounds). The small jobs are handled in the same way as in the previous section. For the large jobs, we will assume that $x_i \leq 1 + \alpha$ for a small $\alpha > 0$. The reason is that if a job with $x_i > 1$ is "on" then it overflows the knapsack capacity regardless of the exact value of $x_i$; Thus, we can set all $x_i > 1 + \alpha$ to $1 + \alpha$. This restricts the range of $x_i$'s to $[t, 1 + \alpha]$. If we round their values with multiplicative error $(1 + \epsilon)$, we get $O(\log 1/t)$ different values, which is a constant since we assume that $p$ is a constant.

Consider now the values of $p_i$'s. Clearly $p_i \leq 1$. Moreover, we can by default include in our solution all jobs with $p_i \leq \epsilon p/n$, since the probability than any of them occurs is less than $\epsilon$. Thus for the remaining jobs we can assume $p_i \geq \epsilon p/n$. Similarily as before, we round all probabilities to the nearest smaller power number of the form $(1 + \delta)^i \cdot \epsilon p/n$ (we will choose $\delta = \Omega(\epsilon^{O(1)} p)$ later). This gives $O((\log \frac{n}{\epsilon p})/\delta)$ classes of probabilities.

As before, for given class of $p_i$ and $x_i$ we need to "guess" the number of jobs from this class which we include in the solution. Since we need to know only approximate number, there are $O(\log n/\epsilon)$ possible approximate values per class pair. Therefore, the total number of possibilities to explore is $(\log n/\epsilon)^{\log t \cdot (\log \frac{n}{\epsilon p})/\delta}$. Notice that if $p$ is constant then $t$ is constant as well. In this case the running time is $n^{O(\log \log n)}$, assuming constant $\epsilon$.

Now we need to justify why rounding $p_i$'s with relative error $(1 + \delta)$ is sufficient. To this end, observe first that we can assume $p \leq 1/(1 + \epsilon)$, since otherwise $p(1 + \epsilon) \geq 1$ and so any solution is feasible. From this and the fact that $x_i \in [0, 1 + \alpha]$, we conclude (by applying standard tail inequalities) that for any set of variables $P$ for which $\mathbf{Pr}\left[\sum_{X_i \in P} X_i \geq 1\right] \leq p \leq 1/(1 + \epsilon)$ we also have $E[\sum_{X_i \in P} X_i] = \sum_{X_i \in P} p_i x_i \leq c \cdot \log 1/\epsilon$, for some constant $c > 0$.

Now we use the following Lemma.

**Lemma 4.7** *Let $X_1 \ldots X_k$ be independent Bernoulli variables such that $\mathbf{Pr}[X_i = x_i] = p_i$, $x_i \in [0, 1]$ and $E[\sum_i X_i] \leq T$. Then for any other sequence of independent Bernoulli variables $X'_1 \ldots X'_k$ such that $\Pr[X'_i = x_i] = p'_i \leq p_i(1 + \delta)$ we have*

$$\mathbf{Pr}\left[\sum_i X'_i \geq 1 + \epsilon\right] \leq \mathbf{Pr}\left[\sum_i X_i \geq 1\right] + \delta T/\epsilon$$

**Proof:** We represent each $X'_i$ as a sum of $X_i$ and $X''_i$, where $X''_i$ are independent Bernoulli variables s.t. $\mathbf{Pr}[X''_i = x_i | X_i = 0] = \frac{p'_i - p_i}{1 - p_i}$ and $\mathbf{Pr}[X''_i = x_i | X_i = 1] = 0$. Notice that $\mathbf{Pr}[X''_i = x_i] = p'_i - p_i \leq \delta p_i$ and that the distributions of $X_i + X''_i$ and $X'_i$ are the same. Now we can write

$$\mathbf{Pr}\left[\sum_i X'_i \geq 1 + \epsilon\right] \leq \mathbf{Pr}\left[\sum_i X_i \geq 1\right] + \mathbf{Pr}\left[\sum_i X''_i \geq \epsilon\right]$$

where the latter probability can be bounded using Markov inequality by

$$\frac{E[\sum_i X''_i]}{\epsilon} \leq \delta T/\epsilon$$

□

Therefore, having $T \leq c \log 1/\epsilon$ and aiming at $\delta T/\epsilon \leq \epsilon p$, we need to set $\delta = \epsilon^2 p / c \log 1/\epsilon$.

## 4.3 Stochastic Bin Packing with Exponential Jobs

In this section we restrict our attention the case where $X_i$ is an exponential random variable with mean $\lambda_i$. The analysis for the stochastic knapsack problem can be extended directly to the bin packing problem to obtain an algorithm such that given any fixed constant $\epsilon$, the following holds:

1. The algorithm partitions the random variables into $k \leq k^*$ sets $S_1, \ldots, S_k$, where $k^*$ is the optimum solution to the problem instance,

2. For each $1 \leq j \leq k$, $\mathbf{Pr}\left[\sum_{i:X_i \in S_j} X_i > 1 + \epsilon\right] \leq p(1 + \epsilon)$, and

3. The running time of the algorithm is polynomial in $n$ and $\log 1/p$.

The details are omitted from this version.

A QPTAS for Stochastic Bin Packing with Bernoulli jobs can similarly be obtained. Chekuri *et al*. [3] recently pointed out that the techniques used in their work on the multiple knapsack problem can be used to obtain a PTAS for Stochastic Knapsack with Bernoulli jobs.

## Acknowledgements

We would like to thank Chandra Chekuri for several helpful discussions and also for information about the recent work by Chekuri *et al*. [3].

## References

[1] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, 1992.

[2] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[3] C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. *Manuscript*, March 1999.

[4] R. Graham. Bounds for certain multiprocessing anomalies. *Bell System Tech. J.*, 45:1563–81, 1966.

[5] D. Hochbaum and D. Shmoys. Using dual approximation algorithms for scheduling problems: practical and theoretixal results. *JACM*, 34(1):144–62, 1987.

[6] O. Ibarra and C. Kim. Fast approximation algorithms for the knapsack and sum of subsets problems. *JACM*, 22:463–468, 1975.

[7] J. Kleinberg, Y. Rabani, and E. Tardos. Allocating bandwidth for bursty connections. *Proc. 29th ACM Symposium on Theory of Computing*, 1997.

[8] E. Lawler. Fast approximation algorithms for knapsack problems. *Math. of Op. res.*, 4:339–356, 1979.

[9] T. Lehtonen. Scheduling jobs with exponential processing times on parallel machines. *J. App. Prob.*, 25:752–62, 1988.

[10] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[11] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, 1995.

[12] R.R. Weber. Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flowtime. *J. Appl. Prob.*, 19, 1982.

[13] G. Weiss. Approximation results in parallel machines stochastic scheduling. *Ann. of Op. res.*, 26:195–242, 1990.

[14] G. Weiss. Turnpike optimality of smith's rule in parallel machines stochastic scheduling. *Math. of Op. res.*, 17:255–270, 1992.

[15] G. Weiss. A tutorial in stochastic scheduling. In P. Chretienne, E.G. Coffman, J.K. Lenstra, and Z. Liu, editors, *Scheduling theory and its applications*. 1995.