2007 Conference on Analysis of Algorithms, AofA 07 DMTCS proc. **AH**, 2007, 519–532

Online Bandwidth packing with symmetric distribution

Marc Lelarge

ENS-INRIA, 45 rue d'Ulm, 75005 Paris

received 26 Feb 2007, *revised 19th January 2008*

We consider the following stochastic bin packing process: the items arrive continuously over time to a server and are packed into bins of unit size according to an online algorithm. The unpacked items form a queue. The items have random sizes with symmetric distribution.

Our first contribution identifies some monotonicity properties of the queueing system that allow to derive bounds on the queue size for First Fit and Best Fit algorithms. As a direct application, we show how to compute the stability region under very general conditions on the input process.

Our second contribution is a study of the queueing system under heavy load. We show how the monotonicity properties allow one to derive bounds for the speed at which the stationary queue length tends to infinity when the load approaches one. In the case of Best Fit, these bounds are tight. Our analysis shows connections between our dynamic model, average-case results on the classical bin packing problem and planar matching problems.

Keywords: Bin packing, stability analysis, heavy traffic, average-case analysis, queueing networks.

1 Introduction

Consider the following bin packing process where items arrive continuously over time to a server and have random sizes in $[0, 1]$. The server contains an infinite number of bins of unit capacity that leaves the system at integer time $t = 0, 1, \ldots$ Bin number t leaves the system at time $t + 1$ with items in the queue that arrived in the system before time $t + 1$, i.e. the t-th bin is packed with items present in the queue at time t and any item that arrive in the t-th slot. The item that were not packed form a queue. To specify completely the dynamic of the system, one has to describe the algorithm for packing the bins. For a given algorithm A, we denote by Q_A the corresponding queue size, i.e. the sum of the sizes of items present in the queue. We restrict ourselves to algorithms that allow to pack a bin with information on the items that are present in the queue and without any information about the items that will arrive in the future.

This general model has been analyzed under various assumptions motivated by bandwidth allocation scheme. For convenience, bin packing terminology shall be used and we refer to the papers cited below for more details on the underlying communication problem. In [13], Kipnis and Robert consider a FIFO scheduling policy and obtain a complete probabilistic distributional solution of the system. In [5], Coffman, Feldman, Kahale and Poonen consider the 'Best Fit' policy for which the largest items departs first.

In the case of discrete distribution for the size of the items, their stability result is extended in the work of Coffman and Stolyar [8] (for symmetric distribution) and Gamarnik [11] (for general distribution). In the most general case, the stability checking algorithm is based on constructing a linear Lyapunov function on the queue length vector process. Although the algorithm is guaranteed to terminate in finite time, the computation time grows exponentially in the number of possible item sizes. In [8], the stability region is derived for the First Fit policy for symmetric discrete distributions and Dantzer and Robert [9] derive also stability characterization when there are only 3 possible item sizes.

We should stress that there is a fundamental difference between the 'Best Fit' and the First Fit policy and we now explain our use of inverted commas. In the classical one-dimensional bin packing problem, an important distinction is made between offline algorithms and online algorithms [10]. An online algorithm for bin packing is one that packs each item as soon as it is received. Clearly in our case, we can use such an online algorithm for our queueing system. In particular, we will show that using the standard First Fit algorithm (as described in [8] or [9]) or the Best Fit algorithm have different implications for the queue length. However 'Best Fit' policy as described in [8], [11] does NOT correspond to the use of the online algorithm Best Fit to pack the bins of the queue.

Our first contribution gives a method to compute an upper bound on the queue size of the queueing system. This method is generic in the sense that it does not depend on the stochastic assumptions and is valid for both First Fit and Best Fit. As an application, we show that it allows to derive the stability region for symmetric distribution (discrete or continuous) for stationary and ergodic inputs: the average load being less than one is a necessary and sufficient condition for the stability of algorithms Best Fit and First Fit. To the best of our knowledge this result was only known for First Fit and for discrete distributions of the items.

Our second contribution is a study of the queueing system under heavy load (or heavy traffic). Similar questions have been addressed in [18] for a policy that was not online and required the knowledge of the rate of arrivals. Moreover only bounds on its performance were derived. Our work allows to derive the speed at which the average queue length tends to infinity when the load approaches one for Best Fit algorithm. Our analysis shows connections between our dynamic model, average-case results on the classical bin packing problem [7] and planar matching problems. Moreover it allows to compare the performance of algorithms which have the same stability region.

2 Results

We now give the stochastic assumptions made on the model presented in previous section. A summary of the notation is given at the end of this section. We assume that the arrival process $N = \{T_n\}$ is stationary and ergodic [2]: the n-th item arrives at time $T_n \in \mathbb{R}$ and has size s_n. We assume that the sequence $\{s_n\}$ is a sequence of independent identically distributed (i.i.d) random variables with $\mathbb{P}(s_1 \in [0,1]) = 1$ and for any $a < b \leq 1/2$, we assume that $\mathbb{P}(s_1 \in (a,b)) = \mathbb{P}(s_1 \in (1-b, 1-a))$, i.e. the distribution is symmetric. This property implies that $\mathbb{E}[s_1] = 1/2$. We denote by 2λ the intensity of the input point process, i.e. $2\lambda = \lim_{t\to\infty} \frac{N(0,t)}{t}$, where we used the standard notation $N(0,t) = \sum_n \mathbb{1}(T_n \in (0,t))$. In this case, the average load is $2\lambda\mathbb{E}[s_1] = \lambda$.

2.1 Stability

Clearly $\lambda \leq 1$ is a necessary condition for the stability of our queueing system regardless of the algorithm used to pack the bins. The first natural question is: given an algorithm A is the corresponding queueing system (called A-system) stable for any $\lambda < 1$? The class of online algorithms if of particular interest: their implementation is easy but their performance is lower than offline algorithm. For the First Fit policy, result of [8] answers positively to previous question when items have discrete size (and under stronger assumption on the input process). Our first theorem extends this result and deals with both Best Fit (BF) and First Fit (FF). In the BF algorithm, each item is placed into the fullest bin in which it fits at the time of arrival. In the FF algorithm, the bins are kept in order. As each item arrives, it is placed into the first bin in which it fits. For both system, at the end of each time slot, the oldest bin of the queue (i.e. the first bin in which an item had been packed) leaves the system.

Theorem 2.1 *Under previous assumptions, the condition* $\lambda < 1$ *is sufficient to ensure the stability of the queueing system with First Fit or Best Fit policy.*

To prove this result (in Section 3) we derive an upper bound for the stationary queue length which corresponds to the stationary workload of a single server $G/G/1$ queue. This technique was introduced by Baccelli and Foss in [3]. They proved that if a certain monotonicity property is satisfied, then there is a generic $G/G/1$ upper bound which is built by making batches of arrivals. Then under the standard condition $\lambda < 1$, we can chose a size of batch sufficiently large in order to make this system stable. In our case, FF and BF are known to behave badly in term of monotonicity and standard restrictions are introduced: MFF and MBF to prevent these anomalies. Recall that Matching Best Fit (MBF) (resp. Matching First Fit (MFF)) behaves exactly like BF (resp. FF), except that MBF (resp. MFF) may not put an item into a bin containing any item of size less than $1/2$. Our first step is then to prove that replacing FF and BF by MFF and MBF can only increase the queue length process. Then we prove that the monotonicity property is satisfied for this new system so that we can apply [3].

2.2 Heavy Load Asymptotics

In view of Theorem 2.1, we cannot distinguish between the FF-system and the BF-system in term of stability region. The main motivation for the rest of our work is to measure the performance of each system.

From now on, we assume that the arrival process is Poisson of intensity 2λ. We do not think that this assumption is crucial but it makes the model simpler. Also instead of looking at the queue length Q in term of the sum of the sizes of the items present in the queue, we consider the variable Z that counts the number of bins actually used in the queue. Clearly $Q \leq Z$ and Z is in a sense a more intrinsic variable since it counts the load of the system in its 'own' unit: if there were no more arrivals then Z would be exactly the time to empty the system.

Theorem 2.2 *Assume that item size is uniformly distributed on* $[0,1]$. *Then we have as* $\lambda \to 1$,

$$\mathbb{E}[Z_{FF}] = \Omega\left(\frac{1}{(1-\lambda)^2}\right), \textit{ and, } \mathbb{E}[Z_{BF}] = \Theta\left(\frac{-\log^{3/2}(1-\lambda)}{(1-\lambda)}\right).$$

This theorem (proved in Section 4) shows that the intuition that the BF-system performs better than the FF-system is correct: when the load is critical, the queue builds up much faster in the FF-system than in the BF-system.

The proof of the lower bounds for Theorem 2.2 follows closely the argument for the lower bound given in [19] for the standard bin packing problem. For the proof of the upper bound, we only deal with the case of MBF which allows us to derive the right asymptotics for BF. We use the standard connection with matching problem [12] but we need to introduce some new constraints on this matching. In our queueing model, a small item cannot be matched to a big item if this one already left the queue at the arrival time of the small item. This mechanism forbids matching between items if the difference of arrival times is large compared to the queue length. Hence we introduce a new system that behaves as the original system when the queue length is small but when the queue length overtake a threshold, we couple our system with a constraint matching. We then use standard concentration results of [1] or [21], to chose properly the threshold and the constraints in order to get the desired upper bound.

To conclude, note that BF is NOT the best online possible bin packing algorithm [20]. Hence there are good chances that other online algorithms can improve the result of order $\Theta\left(\frac{-\log^{3/2}(1-\lambda)}{(1-\lambda)}\right)$. In our case it is easy to adapt the argument of Shor [19] to show the following result (a short proof is given in Section 4.1):

Theorem 2.3 *Under previous assumptions, for any online algorithm A, we have as $\lambda \to 1$,*

$$\mathbb{E}[Z_A] = \Omega\left(\frac{-\log(1-\lambda)}{(1-\lambda)}\right).$$

Note that by considering items of size larger than $1/2$ (which have to be packed in distinct bins), we see that there exists an intrinsic lower bound $\Omega\left(\frac{1}{1-\lambda}\right)$ (which is the standard behavior of a single server queue in heavy traffic). We see that there is a gap between this lower bound and the one obtained in Theorem 2.3. This leaves the following question open: does there exist offline algorithms that perform better than any online algorithm and has a queue size which scales as $\Theta\left(\frac{1}{1-\lambda}\right)$?

2.3 Notation

We use the standard convention for point processes: $N = \{T_n\}$ indexed by $\mathbb{Z}$ with $T_n \leq T_n + 1$ and $T_0 \leq 0 < T_1$. For $n \leq m$, we denote by $N[n,m] = \{T_\ell\}_{n \leq \ell \leq m}$ the $[n,m]$ restriction of N. For $a < b$, we denote by $N(a,b)$ the number of point in the interval (a,b), which is assumed to be finite for any $-\infty < a < b < \infty$. In particular we will use the notation $\mathbb{Z}(a,b)$ to count the number of departure of bins in the interval (a,b). The interval time $[t,t+1]$ with $t \in \mathbb{Z}$, is called the t-th time slot.

For a list of items L, we denote the packing of L using algorithm A by $A(L)$ and the number of bins used by this packing by $\#A(L)$. For an algorithm A and $n \leq m$, let $Q_A[n,m]$ be the sum of the sizes of the items present in the queue at time T_m (just after the arrival of the m-th item) with input restricted to $N[n,m]$ when algorithm A is used to pack the bins and the first bin of the packing is leaving the system at discrete times. We denote by $P_A[n,m]$ the packing of the items in the queue of the system and by $Z_A[n,m] = \#P_A[n,m]$ the number of bins. Of course $Q_A[n,m] \leq Z_A[n,m]$. Note also that $\lceil T_m \rceil + Z_A[n,m]$ is the time at which the system empties when inputs are restricted to $N[n,m]$ and where $\lceil x \rceil$ is the integer such that $x \leq \lceil x \rceil < x+1$.

3 Stability of Online Bin Packing Algorithms

In this section we establish the stability result claimed for BF and FF. Most of the argument is identical for both algorithms. In particular, we use MBF and MFF in order to find an upper bound with some monotonicity properties. To state properties that are true for both BF anf FF, we use the generic notation

A. For example, $\#A(L) \leq \#MA(L)$ means that both $\#BF(L) \leq \#MBF(L)$ and $\#FF(L) \leq \#MFF(L)$ are correct.

3.1 Monotonicity properties

We first deal with algorithms MBF and MFF. It is convenient to introduce virtual items corresponding to deleted items: these items are ignored by any packing algorithm. We take the convention that virtual items have a null size so that the action of deleting an item from a list L is the same as replacing its positive size by a null size. With this convention, we can state our next lemma,

Lemma 3.1 *Let $\{T'_n, s'_n\}$ be a modification of the original input $\{T_n, s_n\}$, obtained by deletion of some items, i.e. for all n, we have $T_n = T'_n$ and either $s'_n = s_n$ or $s'_n = 0$. Let $Z'_{MA}[n,m]$ be the queue size of the MA-system with deleted items. We have $Z'_{MA}[n,m] \leq Z_{MA}[n,m]$.*

Proof: We first recall a result that can be found in [4] for MFF and in [19] for MBF. If L' is a list obtained by removing one item from L, then if we consider any bin with two items in them to be identical, $MA(L)$ and $MA(L')$ differ in at most one bin:

(i) $MA(L)$ can be obtained from $MA(L')$ by replacing a 1-item bin by a full (2-item) bin.

(ii) $MA(L)$ can be obtained from $MA(L')$ by adding a 1-item bin.

In particular, the proofs in [4] and [19] show that if we add an item to two packings related by (i) or (ii), they will still be related by (i) or (ii). In our setting, we need to consider also the case where a bin is removed.

More precisely, we need to prove Lemma 3.1 for one deletion only, the result then follows by an easy induction. To simplify notation, assume that item s_0 arriving at time T_0 is deleted. We have $Z'_{MA}[n,m] = Z_{MA}[n,m]$ for $m \leq -1$ or $n \geq 1$. Consider the case $n \leq 0$. We now prove by induction that for $m \geq 0$, we can relate the packing $P_{MA}[n,m]$ and $P'_{MA}[n,m]$ in one of the following way:

(a) $P_{MA}[n,m]$ can be obtained from $P'_{MA}[n,m]$ by replacing a 1-item bin by a full bin.

(b) $P_{MA}[n,m]$ can be obtained from $P'_{MA}[n,m]$ by adding a (1-item or 2-item) bin.

(c) $P_{MA}[n,m] = P'_{MA}[n,m]$.

Consider two packings related by (a), (b) or (c). From the previous discussion, we see that when adding an item, these packings are still related by (a), (b) or (c). The only case which requires a proof is when one packing is obtained from the other by adding a full (2-item) bin. The added item is packed in the same bin in both packings, so that the two packings are still related by (b). We still consider the two packings and remove the first bin from each of them. If the two packings were related by (a), then if the first bin is different, the resulting packings will be related by (c) and by (a) otherwise. If the two packings were related by (b), then the resulting packings will be related by (b) unless one of the packing was empty and then the resulting packings are both empty and hence related by (c). We must still do the first case of the induction: for $m = 0$, $P_{MA}[n,0]$ and $P'_{MA}[n,0]$ are related by (i) or (ii) defined above which are sub-cases of (a) or (b). □

Lemma 3.2 *For all input $N' = \{T'_k\}$ that satisfies $T'_k \geq T_k$ for all k, we have for all $n \leq m$,*

$$\lceil T_m \rceil + Z_{MA}[n,m] \leq \lceil T'_m \rceil + Z'_{MA}[n,m].$$

Proof: We clearly need to prove the lemma for one delayed arrival time only. Suppose that $T_k \leq \ell < T'_k$ with $\ell \in \mathbb{Z}$. Note in particular that the k-th item is the last one arrived in slot $\ell - 1$ in the original system. Then we have $Z[n,k-1] = Z'[n,k-1]$. If the k-th item does not leave the original system in slot $\ell - 1$, then we have $Z[n,m] = Z'[n,m]$ for all $m \geq k$. Suppose now that the k-th item leaves the original system in slot $\ell - 1$. First if k is the only item leaving the original system, then the system is empty at time ℓ. Hence we have

$$\begin{aligned} \lceil T_k \rceil + Z_{MA}[n,k] = \ell &\leq \lceil T'_k \rceil + Z'_{MA}[n,k], \\ \lceil T_m \rceil + Z_{MA}[n,m] = \lceil T_m \rceil + Z_{MA}[k+1,m] &\leq \lceil T'_m \rceil + Z'_{MA}[n,m] = \lceil T'_m \rceil + Z'_{MA}[k,m], \text{ for } m > k. \end{aligned}$$

This implies that for all $m \geq k$, we have $Z_{MA}[n,m] = Z_{MA}[k+1,m]$ and $Z'_{MA}[n,m] = Z'_{MA}[k,m]$ and the claim follows from Lemma 3.1. Now if k is the second item of the bin leaving the original system, then the first item also leaves the delayed system and the claim still follows from Lemma 3.1. □

3.2 Upper Bound

Lemma 3.2 proves that delaying arrival times of the items to the MA-system results in a delay for the last departure time (when arrivals are finite). This property ensures that the MA-system belongs to the monotone separable framework introduced in [3]. The analysis of the stability of such system is then standard and can be done by considering batches as follows: for $j \leq 0$, all items arrived between $T_{(j-1)L+1}$ and T_{jL} are delayed to the time T_{jL}. All items form the j-th batch are packed into bins after time T_{jL} and after all items from previous batch have already left the system. We denote by $Z^0_{MA}[(j-1)L+1, jL]$ the number of bins needed by MA to pack the items of the j-th batch. Since we only delayed arrival times, the following proposition follows from Lemma 3.2. We refer to [17] (Proposition 4.2) for a proof.

Proposition 3.1 *Let $L > 0$ and $n \geq 0$. We denote $k = \lfloor n/L \rfloor$, then we have:*

$$Z_{MA}[-n,0] \leq Z^0_{MA}[-L+1,0] + \sup_{i=-k}^{-1} \sum_{j=i}^{-1} \left(Z^0_{MA}[(j-1)L+1, jL] - \tau_j \right), \tag{3.1}$$

with $\tau_j = T_{jL} - T_{(j-1)L}$.

Note that the left-hand side of (3.1) is the time of last departure from the original system when arrivals are stopped after time 0 and the right-hand side is the time of last departure for the delayed system with batches described above.

We need an auxiliary lemma to relate the A-system to the MA-system, as follows,

Lemma 3.3 *We have for any $n + 1 \leq m$, $Z_A[n,m] \leq Z_{MA}[n,m]$.*

Proof: Consider the A-system and remove from the arrival process all the items which were packed by the A-system into a bin already containing an item of size less than $1/2$. We remove all items that are packed into bins considered full by MA, so the A-system and MA-system behave exactly in the same manner with this thinned arrival process and the lemma follows directly from Lemma 3.1. □

3.3 Stability Results

Putting together previous results, it is easy to show that $Q_A := \limsup_{n\to\infty} Q_A[-n,0]$ is finite for $A = FF, BF$. The general idea is to choose the size L of the batches sufficiently large so that the right-hand side of (3.1) is finite when the supremum runs over all $i \leq -1$. The condition $\lambda < 1$ is sufficient to ensure the existence of such a L and Theorem 2.1 follows. We refer to [17] for details.

4 Performance of Online Algorithms under Heavy Load

From now on, we assume that the arrival process is a Poisson point process of intensity 2λ and that item sizes are uniformly distributed in $(0,1)$. An easy coupling argument gives:

Lemma 4.1 *If* $\limsup_{n\to-\infty} Z_A[-n,0] < \infty$*, then the limit* $\lim_{n\to\infty} P_A[-n,0] = P_A$ *exists almost surely and is the unique stationary* A*-system. Moreover there are an infinite number of negative times* $t \in \mathbb{Z}$ *and an infinite number of positive times* $t \in \mathbb{N}$ *such that the* A*-system empties at these times.*

Note in particular that results of previous section ensure that $\limsup_{n\to-\infty} Z_A[-n,0] < \infty$ for $A = BF, FF$ as long as $\lambda < 1$. Hence in this case, there exists an unique stationary A-system.

4.1 A General Lower Bound

We give a short proof of Theorem 2.3 that follows the proof of the lower bound for online binpacking (Section 2 in [19]). Several concepts will be useful for the rest of the paper.

We assume that algorithm A is online and that the A-system satisfies the conditions of Lemma 4.1 for all $\lambda < 1$. In particular, we can define the stationary A-system for any $\lambda < 1$.

To convert the list of items arrived at times $\{T_0, \dots T_n\}$ (with $n \geq 0$) to a planar matching problem, we represent the items received by the algorithm by points in a rectangle $[0,n] \times [0,1]$. The y-coordinate will be the size of the item and the x-coordinate will be the index of the item. Note that we use a different convention from the binpacking literature. To the j-th item arriving at time T_j, we associate the point (j, s_j). Next, we label the items larger than $1/2$ with $'+'$ and those smaller than $1/2$ with $'-'$. We then fold the plane about the line $y = 1/2$, so a $+$ point with y-coordinate $s > 1/2$ will be moved so it has y-coordinate $1-s$. We focus on just those items with size in $(1/3, 2/3)$. We join every $+$ point in this range to a $-$ point in this range representing an item packed in the same bin, if there is such an item. This gives a bipartite matching $M[0,n]$ between $+$ and $-$ points. In other words, every couple of items represented by a matching in $M[0,n]$ will leave the stationary A-system in the same bin. For $n > 0$, we denote by $U[0,n]$ the number of unmatched $-$ points in $M[0,n]$ and by $L[0,n]$ the number of items with size larger than $1/2$ arrived on $[T_0, T_n]$. As in [19], $L[0,n] + 1/2U[0,n]$ is a lower bound on the number of bins used to pack items arrived on $[T_0, T_n]$, so that we have for the stationary process,

$$Z_A(T_n) \geq L[0,n] + \frac{U[0,n]}{2} - \lfloor T_n \rfloor.$$

Based on [1], Shor shows in Lemma 1 of [19] that for n sufficiently large, we can find a constant C such that

$$\mathbb{E}\left[\frac{1}{n}\sum_{k=0}^{n} U[0,k]\right] \quad \geq \quad C\sqrt{n \log n}.$$

Hence for sufficiently large values of n, we have for the stationary process:

$$\begin{aligned}\mathbb{E}[Z_A] = \frac{\mathbb{E}\left[\int_{T_0}^{T_n} Z_A(s)ds\right]}{\mathbb{E}[T_n - T_0]} &\geq \frac{C}{2}\sqrt{n\log n} + \frac{\lambda}{n}\sum_{k\leq n}\left(\frac{k}{2} - \frac{k}{2\lambda} - 1\right)\\ &\geq \frac{C}{2}\sqrt{n\log n} + (\lambda - 1)\frac{n-1}{2}.\end{aligned}$$

Take $n \approx \frac{-\log(1-\lambda)}{(1-\lambda)^2}$, to get

$$\mathbb{E}[Z_A] = \Omega\left(\frac{-\log(1-\lambda)}{(1-\lambda)}\right).$$

4.2 Lower bound for Best Fit

Thanks to Lemma 4.1, we know that there exists an unique stationary process Z_{BF}. The idea is to consider this process at a well-chosen deterministic time t. By stationarity we have $\mathbb{E}[Z_{BF}] = \mathbb{E}[Z_{BF}(t)]$. Moreover if $f(t)$ is a lower bound on the number of bins packed with items arrived on the time interval $[0,t]$, we have

$$\mathbb{E}[Z_{BF}(t)] \geq \mathbb{E}[f(t)] - \lfloor t \rfloor, \text{ so that, } \mathbb{E}[Z_{BF}] \geq \sup_{t\geq 0}\mathbb{E}[f(t)] - \lfloor t \rfloor.$$

We show how to adapt the argument of Section 3b in [19] in order to get a good lower bound $f(t)$. We will consider the stationary BF-system and what happens to the items that arrive after time 0 and with size between $1/3$ and $2/3$. We will call items (arrived after time 0) with size between $1/3$ and $1/2$ s-items and items with size between $1/2$ and $2/3$ b-items. We will call a bin with two s-items a $\left[{s \atop s}\right]$ bin and denote the number of these bins by $\left|{s \atop s}\right|$ as in [19]. There are five types of bins: $\left[{s \atop s}\right]$, $\left[{s \atop b}\right]$, $\left[{b \atop s}\right]$, $\left[{\ \atop s}\right]$ and $\left[{\ \atop b}\right]$. Note that we restrict our attention to items arrived after time 0. In particular, any of these bins might contain an item that arrived before time 0. This would correspond to the case where the s- or b-items are packed in bins already existing at time 0.

With these notations, we can follow the argument of Section 3b [19] even if our packing is NOT the same as the one obtained with Best Fit applied on items arrived after time 0. There is an important feature of our system which shows that Lemma 4 in [19] is still valid in our framework.

Lemma 4.2 *For $A = BF$ or FF, we have $P_A[n,m] = A(L[n,m])$ where $L[n,m]$ is the list of items arrived on the time interval $[T_n, T_m]$ from which we deleted the items departed from the A-queueing system.*

Proof: This lemma follows from the following condition satisfied by $A = BF$ or $A = FF$:
Suppose B is the set of items contained in Bin j of the packing of L obtained with A. Then if we delete the items in B from L and pack the resulting list by A, the resulting packing will be identical to that obtained by deleting Bin j from the original packing. □We look at the system at time t and denote by n

the number of items arrived on the interval $(0,t)$. Then following the argument of the proof of the lower bound for Best Fit, we get that there are $\Omega(\sqrt{n}\log^{3/4} n)$ bins containing no items larger than $1/2$ with probability at least $1 - 1/n$. This shows that the number of bins used is $f(n) = n/2 + \Omega(\sqrt{n}\log^{3/4} n)$ with probability at least $1 - 1/n$. Hence we have for any $t > 0$, with $n_t = \lfloor 2\lambda t \rfloor$,

$$\begin{aligned}
\mathbb{E}[Z_{BF}(t)] &\geq \mathbb{E}[f(N(0,t))] - \lfloor t \rfloor \\
&\geq \mathbb{E}[f(n_t)] + O(\sqrt{n_t}) - t \\
&\geq \left(1 - \frac{1}{n_t}\right)\left(n_t/2 + \Omega(\sqrt{n_t}\log^{3/4} n_t)\right) - t,
\end{aligned}$$

where the second inequality follows from a bound for Poisson random variables see Lemma 2.1 [6]. Then, as explained above, optimizing the right-hand side by taking $t \sim \frac{-\log^{3/2}(1-\lambda)}{(1-\lambda)^2}$, directly gives

$$\mathbb{E}[Z_{BF}] = \Omega\left(\frac{-\log^{3/2}(1-\lambda)}{(1-\lambda)}\right).$$

4.3 Lower bound for First Fit

We show how we can adapt the argument of Section 4b of [19]. We call the subsequence of bins that have more empty space than all preceding bins the greedy decreasing subsequence of bins. An item that is packed by FF will go into one of the bins of this subsequence. Let Q_j be the tail of the greedy decreasing subsequence starting with the first bin less than $2/3$ in the packing of $P_{FF}(-\infty, j]$. All the bins in Q_j except possibly for the first one contain only one item. As shown in [19], if $|Q_j| \leq k$, then there is a probability $7/12$ that an item will leave a space $1/(12k)$ or greater in a bin. Hence if Q_j is never longer than $n^{1/3}$ for $j \leq n$, then every item had a $7/12$ probability of leaving $(1/12)n^{-1/3}$ or more empty space in the bin it entered. So we have $\Omega(n^{2/3})$ wasted space in the packing of all the items arrived on $[0, T_n]$, i.e. the packing obtained by concatenation of all bins that left the system at time T_n and the bins of $P_{FF}(-\infty, n]$. On the other hand, if the subsequence Q_j were longer than $n^{1/3}$ at some time $j \leq n$, we would have that at this time $\Omega(n^{2/3})$ bins with at least $1/6$ empty space. To see this, note that the argument of Shor using the longest increasing subsequence is still valid. As in [19], one can show that with high probability, we still have $\Omega(n^{2/3})$ wasted space for $P_{FF}(-\infty, n]$. Hence we proved that

$$\lfloor T_n \rfloor + Z_{FF} \geq \frac{n}{2} + \Omega(n^{2/3}), \quad \text{with high probability.}$$

Hence taking the expectation and optimizing in n, we get

$$\mathbb{E}\left[Z_{FF}\right] \geq \sup_{n \geq 0}\left\{\frac{n}{2} + \Omega(n^{2/3}) - \frac{n}{2\lambda}\right\}.$$

Choosing n of order $\frac{1}{(1-\lambda)^3}$, it is easy to get as $\lambda \to 1$,

$$\mathbb{E}[Z_{FF}] = \Omega\left(\frac{1}{(1-\lambda)^2}\right).$$

4.4 Upper bound for Best Fit

As shown in Section 3, we need to consider the MBF-system only and we first introduce a modification of this system. For $K > 0$ fixed, we consider the (K, MBF)-system that behaves exactly like the MBF-system except that if a new item arrives at time T_n and if this item would have been packed with an item arrived before T_{n-K} in the original system, then this item is packed in a new bin (and will remain the only item of its bin). We denote by Z^K_{MBF} the corresponding queue length process (measured in number of non-empty bins). One can relate the (K, MBF)-system to a standard MBF-system with delayed arrivals and then apply Lemma 3.2, to get:

Lemma 4.3 *We have for any* $n \le m$, $Z_{MBF}[n, m] \le Z^K_{MBF}[n, m]$.

Now consider the rectangle $[n, m] \times [0, 1/2]$. For $\ell \in [n, m]$, we represent the ℓ-th item by a $+$ located at $(\ell, 1 - s_\ell)$ if $s_\ell > 1/2$ and by a $-$ located at (ℓ, s_ℓ) otherwise. An up-left matching is a one-to-one matching of pluses and minuses such that every plus is either unmatched or is matched to a single minus that lies below and to the right of the plus. It is well-known that MBF produces an optimal up-left matching (in the sense that the number of pairs is maximized). We denote by $M^K[n, m]$ the matching obtained from this matching on the list of items arrived during the time interval $[T_n, T_m]$ where pairs (a, b) with x-coordinate $n_a < n_b$ are cut if $n_b - n_a > K$. Following [15], on can show that (see [16] for a proof)

Lemma 4.4 *With high probability (at least* $1 - 1/L^\alpha$ *for any* $\alpha \ge 1$*) we have,*

$$M^K[1, L] = L/2 + \Theta\left(\sqrt{L}\log^{3/4} L\right),$$

as long as $K = \Omega\left(\sqrt{L}\log^{3/4} L\right)$.

We now show how these results allow to get an upper bound for the scaling of the average queue size of the BF-system.

Lemma 4.5 *For* K *and* L *defined as above and* λ *sufficiently close to one, we have*

$$\begin{aligned} Z_{MBF}(-\infty, L] \quad \le \quad & K + \max\left\{Z_{MBF}(-\infty, 0] + M^K[1, L] - \mathbb{Z}(T_0, T_L)\right. \\ & \left.\sup_{0 \le \ell \le L}\left\{M^K[\ell, L] - \mathbb{Z}(T_\ell, T_L)\right\}\right\}. \end{aligned} \tag{4.1}$$

Proof: Recall that $Z_{MBF}(-n, 0] \nearrow Z_{MBF}(-\infty, L] < \infty$ as $n \to \infty$ thanks to results of Section 3. If $Z_{MBF}(-\infty, L] \le K$, then (4.1) is obvious. Assume $Z_{MBF}(-\infty, L] > K$ and let $\tau = \sup\{\ell,\ Z_{MBF}(-\infty, \ell] \le K\}$. Define the modified system $\tilde{Z}^K_{MBF}(-\infty, \ell]$ that behaves exactly as $Z_{MBF}(-\infty, \ell]$ for $\ell \le \tau$ and for $\ell > \tau$ behaves like the (K, MBF)-system except that bins present in the queue at time T_τ are considered as closed for new incoming items. We have clearly $K \le Z_{MBF}(-\infty, \ell] \le \tilde{Z}^K_{MBF}(-\infty, \ell]$ for all $\ell \ge \tau + 1$. Hence we have

$$Z_{MBF}(-\infty, L] \quad \le \quad Z_{MBF}(-\infty, \tau] + M^K[\tau + 1, L] - \mathbb{Z}(T_\tau, T_L).$$

Since $Z_{MBF}(-\infty, \tau] \le K$, note that if $\tau > 0$, there is nothing to prove, if $\tau < 0$, then consider the modified system from time 1, and we get similarly

$$Z_{MBF}(-\infty, L] \quad \le \quad Z_{MBF}(-\infty, 0] + M^K[1, L] - \mathbb{Z}(T_0, T_L).$$

□From a recursion like (4.1) it is quite standard to derive an upper bound on the mean of $Z_{MBF}(-\infty, L]$ see Chapter 5 in [14] we take squares of (4.1) and then expectation. If we write $Y = K + M^K[1, L] - \mathbb{Z}(T_0, T_L)$ and $X = K + \sup_{0\leq \ell \leq L}\left\{M^K[\ell, L] - \mathbb{Z}(T_\ell, T_L)\right\}$, we obtain

$$-2\mathbb{E}\left[Z_{MBF}(-\infty, 0]\right]\mathbb{E}\left[Y\right] \leq \mathbb{E}\left[Y^2\right] + \mathbb{E}\left[X^2\right]$$

With the following choices of L and K,

$$L = \frac{-4\log^{3/2}(1-\lambda)}{(1-\lambda)^2} \text{ and, } K = \sqrt{L}\log^{3/4} L,$$

we have

$$-\mathbb{E}\left[Y\right] = \Omega\left(\frac{\log^{3/2}(1-\lambda)}{1-\lambda}\right).$$

It is also easy to derive from Lemma 4.4 that

$$\mathbb{E}\left[Y^2\right] \text{ and, } \mathbb{E}\left[X^2\right] = O\left(\frac{\log^{3/2}(1-\lambda)}{1-\lambda}\right)^2,$$

and the asymptotics for BF in Theorem 2.2 easily follows.

Acknowledgements

This work was motivated by a presentaion given by Pr. Tsitsiklis during the Conference on Stochastic Networks (UIUC 2006).

References

[1] M. Ajtai, J. Komlós, and G. Tusnády. On optimal matchings. *Combinatorica*, 4(4):259–264, 1984.

[2] F. Baccelli and P. Brémaud. *Elements of queueing theory*, volume 26 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, second edition, 2003. Palm martingale calculus and stochastic recurrences, Stochastic Modelling and Applied Probability.

[3] F. Baccelli and S. Foss. On the saturation rule for the stability of queues. *J. Appl. Probab.*, 32(2):494–507, 1995.

[4] J. L. Bentley, D. S. Johnson, F. T. Leighton, C. C. McGeoch, and L. A. McGeoch. Some unexpected expected behavior results for bin packing. In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 279–288, New York, NY, USA, 1984. ACM Press.

[5] E. G. Coffman, Jr., A. Feldmann, N. Kahale, and B. Poonen. Computing call admission capacities in linear networks. *Probab. Engrg. Inform. Sci.*, 13(4):387–406, 1999.

[6] E. G. Coffman, Jr., D. S. Johnson, P. W. Shor, and R. R. Weber. Bin packing with discrete item sizes. II. Tight bounds on first fit. *Random Structures Algorithms*, 10(1-2):69–101, 1997. Average-case analysis of algorithms (Dagstuhl, 1995).

[7] E. G. Coffman, Jr. and G. S. Lueker. *Probabilistic analysis of packing and partitioning algorithms*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1991. , A Wiley-Interscience Publication.

[8] E. G. Coffman, Jr. and A. L. Stolyar. Bandwidth packing. *Algorithmica*, 29(1-2):70–88, 2001. Average-case analysis of algorithms (Princeton, NJ, 1998).

[9] J.-F. Dantzer and P. Robert. Fluid limits of string valued Markov processes. *Ann. Appl. Probab.*, 12(3):860–889, 2002.

[10] J. E. G. Coffman, M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: a survey. pages 46–93, 1997.

[11] D. Gamarnik. Stochastic bandwidth packing process: stability conditons via Lyapunov function technique. *Queueing Syst.*, 48(3-4):339–363, 2004.

[12] R. M. Karp, M. Luby, and A. Marchetti-Spaccamela. A probabilistic analysis of multidimensional bin packing problems. In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 289–298, New York, NY, USA, 1984. ACM Press.

[13] C. Kipnis and P. Robert. A dynamic storage process. *Stochastic Process. Appl.*, 34(1):155–169, 1990.

[14] L. Kleinrock. *Queueing Systems, Volume 1: Theory*. John Wiley & Sons, 1975.

[15] T. Leighton and P. Shor. Tight bounds for minimax grid matching with applications to the average case analysis of algorithms. *Combinatorica*, 9(2):161–187, 1989.

[16] M. Lelarge. Optimal bandwidth packing with symmetric distribution. in preparation.

[17] M. Lelarge. Tail asymptotics for monotone-separable networks. *J. Appl. Probab.*, 44(2), 2007.

[18] D. Shah and J. N. Tstsiklis. Bin packing with queues, 2006. http://web.mit.edu/jnt/www/publ.html.

[19] P. W. Shor. The average-case analysis of some on-line algorithms for bin packing. *Combinatorica*, 6(2):179–200, 1986. Theory of computing (Singer Island, Fla., 1984).

[20] P. W. Shor. How to pack better than best fit: tight bounds for average-case on-line bin packing. In *Proceedings of the 32nd annual symposium on Foundations of computer science*, pages 752–759, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[21] M. Talagrand. Matching theorems and empirical discrepancy computations using majorizing measures. *J. Amer. Math. Soc.*, 7(2):455–537, 1994.