Queueing Delay in Bin-Packing and other Stochastic Combinatorial Problems

John N. Tsitsiklis

(with Devavrat Shah)

Massachusetts Institute of Technology

2006 Stochastic Networks Conference
U. Illinois at Urbana-Champaign

June 2006

Outline

- Probabilistic bin-packing, TSP, and TRP
- Queueing variants
- Batching heuristic
- Tightness of batching in TSP and TRP
- A bin-packing policy
- Lower bounds in bin-packing

Static Probabilistic Bin-Packing (Rhee et al.)

- n items
- Sizes $X_1, \ldots, X_n$; i.i.d. $X_i \sim U[0,1]$
- N_n: min # of unit bins needed

$$\mathbb{E}[N_n] = \frac{n}{2} + \Theta(\sqrt{n})$$

N_n is "concentrated" around its mean

On-Line Probabilistic Bin-Packing (Shor)

- Place each item as soon as it arrives

$$N_n \geq \frac{n}{2} + \Omega\left(\sqrt{n}\,\log^{1/2} n\right), \qquad \text{w.h.p.}$$

- Best-Fit: (place item so that resulting bin is as full as possible)

$$N_n \leq \frac{n}{2} + O\left(\sqrt{n}\,\log^{3/4} n\right), \qquad \text{w.h.p.}$$

Probabilistic TSP

- n points/cities, i.i.d. locations $X_i \in [0,1]^2$
- L_n: length of shortest TSP tour

$$\frac{L_n}{\sqrt{n}} \to c, \text{ a.s.} \qquad \frac{\mathbb{E}[L_n]}{\sqrt{n}} \to c; \qquad \mathrm{Var}(L_n) = O(1)$$

- Proof of $\mathbb{E}[L_n] \sim \sqrt{n}$:

$$L_n \leq 2\sqrt{n} \quad \text{(space-filling curve)}$$

$$\mathbb{E}\Big[\min_j \|X_i - X_j\|\Big] = \Omega(1/\sqrt{n})$$

Probabilistic TRP

- Each city i has a service time S_i
- S_i: i.i.d.; $\mathbb{E}[S_i] = 1$ $\mathbb{E}[S_i^2] < \infty$
- Time to serve all cities: $T_n \sim n + \Theta(\sqrt{n})$

Online Problems with Queueing

Demand:

TSP/TRP:	Cities arrive as Poisson(λ)
Bin-Packing:	Items arrive as Poisson(2λ)

Service:

TSP/TRP:	Move at unit speed, and serve at unit rate
Bin-Packing:	Each time unit: 1 available bin (use it or lose it)

Performance Measures:

Throughput:	maximum λ for which system is “stable”
Queue Size:	# of unserved items/cities
Delay:	Time from arrival until service completion

Batching Heuristic

Collect items/cities over t time steps, and "process" as one job (G/D/1 queue)

Problem	**Arriving work**	**Variance**
TSP	$\sqrt{\lambda t}$	1
TRP	$\lambda t + \sqrt{\lambda t}$	λt
Bin-packing	$\lambda t + \sqrt{\lambda t}$	λt

Stability

Problem	**Condition**	**Throughput**	**Batching time**
TSP	$\sqrt{\lambda t} < t$	$\lambda < \infty$	λ
TRP	$\lambda t + \sqrt{\lambda t} < t$	$\lambda < 1$	$(1 - \lambda)^{-2}$
Bin-packing	$\lambda t + \sqrt{\lambda t} < t$	$\lambda < 1$	$(1 - \lambda)^{-2}$

Delay (use Kingman's bound for G/D/1 queues)

TSP: λ TRP, BP: $(1 - \lambda)^{-2}$

$\Omega(\lambda)$ Delay for TSP

- Start "empty"
- Expected "work" arrived by time $t = \lambda/4$ is: $\geq 0.7\sqrt{\lambda t} = 0.35\lambda$
- Expected remaining work $\geq 0.1\lambda$
- Work with n items $\leq 2\sqrt{n}$
- Expected # of items = $\Omega(\lambda^2)$
- Little's law: Expected delay = $\Omega(\lambda)$

$\Omega((1-\lambda)^{-2})$ Delay for TRP (Bertsimas & Van Ryzin)

- "Typical" item i^* arrives
- Finds K items in queue
- Delay D until it's service starts
- M items arrive during the delay

$$\begin{aligned}
1-\lambda &> \text{expected travel time from item served just before } i^* \\
&\geq \text{expected nearest distance to } 1+K+M \text{ points} \\
&\geq c\mathbb{E}\left[\frac{1}{\sqrt{1+K+M}}\right] \\
&\geq c\frac{1}{\sqrt{\mathbb{E}[1+K+M]}} \\
&\sim c'\frac{1}{\sqrt{\mathbb{E}[D]}}
\end{aligned}$$

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

Tightness of Batching?

$$\Theta(\sqrt{n}) \quad \text{"waste"} \qquad \Longleftrightarrow \qquad \Omega\left(\frac{1}{(1-\lambda)^2}\right) \quad \text{delay ?}$$

- TRP: lower bound rested on
  expected number of items needed to find a good match

- Bin-Packing: enough items need to arrive to find a match within $1 - \lambda$

- Only need to wait for $O\left(\frac{1}{1-\lambda}\right)$ items (and also time)

Naive policy: Just look for a "good match"

- Break $[0, 1]$ into $2m + 1$ equal intervals
- One queue for items in intervals 1 and $2m$;
  same for 2 and $2m - 1$, etc.
- Allocate $1/(m + 1)$ of the incoming bins to each queue
- Waste $= \Omega(1/m)$;   Stability $\implies m = \Omega((1 - \lambda)^{-1})$
- Each queue has utilization factor $\rho \approx 1 - \lambda$
- Total queue size: $m \cdot \frac{1}{1 - \lambda} = \Omega\left(\frac{1}{(1 - \lambda)^2}\right)$

More general fixed-pair policies?

- $2m + 1$ queues as before; $\quad m \approx \dfrac{1}{1-\lambda}$
- $Q(t) \geq \sum_{i=1}^{m} \max\left\{N_i(t), N_{2m+1-i}(t)\right\} - t$
- $\max\{M, N\} = \frac{M+N}{2} + \frac{|M-N|}{2}$
- $\mathbb{E}[Q(t)] \geq \lambda t + \dfrac{1}{1-\lambda}\sqrt{t(1-\lambda)} - t$
- Consider time $t = (1-\lambda)^{-3}$
- $\mathbb{E}[Q(t)] \geq c\dfrac{1}{(1-\lambda)^2}$

A Better Policy

- Divide bins into types: $0, 1, \ldots, D, U$
- Type i bins can serve simultaneously:
  - one item in interval $1, \ldots, i$
  - one item in interval $m + 1, \ldots, 2m - i$
  - priority to larger items

Interval 9							+
Interval 8	+					+	
7	+	+				+	
6	+	+	+			+	
5	+	+	+	+		+	
4				+	+		
3			+	+	+		
2		+	+	+	+		
1	+	+	+	+	+		
Bin type:	1	2	3	4	D	U	0
Bin Rate:	= arrival rate				$c(1-\lambda)$		

A Better Policy

- Divide bins into types: $0, 1, \ldots, D, U$
- Type i bins can serve simultaneously:
  - one item in interval $1, \ldots, i$
  - one item in interval $m + 1, \ldots, 2m - i$
  - priority to larger items

Interval 9							+
Interval 8	+					+	
7	+	+				+	
6	+	+	+			+	
5	+	+	+	+		+	
4				+	+		
3			+	+	+		
2		+	+	+	+		
1	+	+	+	+	+		
Bin type:	1	2	3	4	D	U	0
Bin Rate:	= arrival rate				$c(1 - \lambda)$		

Two halves decouple!

Some Details

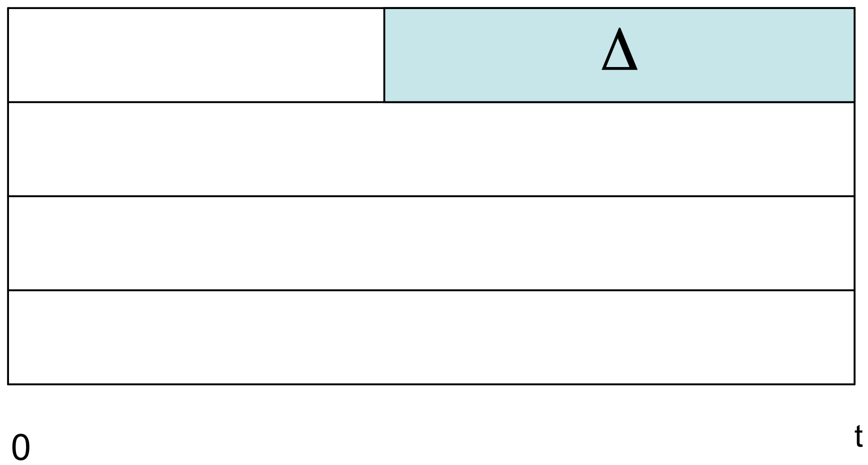
- Can assume bins of each type are arriving regularly, according to prescribed rate

- Can assume bins of each type arrive as Poisson process, at slightly modified rate

- Can assume bin-type is uniform in $[0, m/(2m+1)]$

4				+	+
3			+	+	+
2		+	+	+	+
1	+	+	+	+	+
Bin type:	1	2	3	4	D
Bin Rate:	= arrival rate				$c(1-\lambda)$

Preliminaries: Queue for items of type m

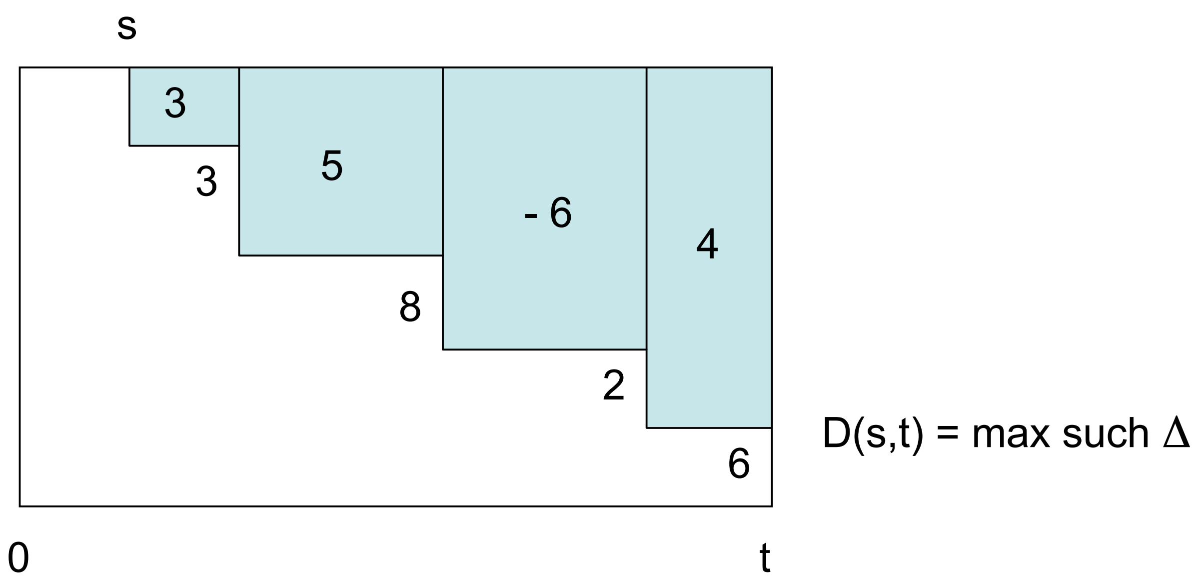


- Δ = (# arrivals) $-$ (available bins)

- $\mathbb{E}[\Delta]$ = $-c \cdot (1 - \lambda) \times$ (time interval)

- $Q_m(t) = \max_{s \in [0,t]} \Delta_m(s,t)$

Queue Dynamics



$$Q(t) \geq \max_{s\in[0,t]} D(s,t)$$

Argument can be reversed:

If $Q(0) = 0$, then $Q(t) = \max_{s\in[0,t]} D(s,t)$

Upper Bound

$$Q(t) \leq \max\left\{ \max_{0 \leq s \leq t} D(s,t),\ \ Q(0) + D(0,t) \right\}$$

- Take squares of both sides (in steady-state)

$$\mathbb{E}[Q] \leq \frac{\mathbb{E}\left[\left(\max_{s \in [0,t]} D(s,t)\right)^2\right]}{|\mathbb{E}[D(0,t)]|}$$

Lemma: (based on Leighton & Shor)

$$\mathbb{E}[D(0,t)] \leq -c_1(1-\lambda)t + c_2\sqrt{t}\log^{3/4} t$$

$$\mathbb{E}\left[\left(\max_{s \in [0,t]} D(s,t)\right)^2\right] \leq c_3 t \log^{3/2} t$$

Upper Bound (cont.)

$$\mathbb{E}[Q(t)] \le \frac{\mathbb{E}\left[\left(\max_{s\in[0,t]} D(s,t)\right)^2\right]}{|\mathbb{E}[D(0,t)]|}$$

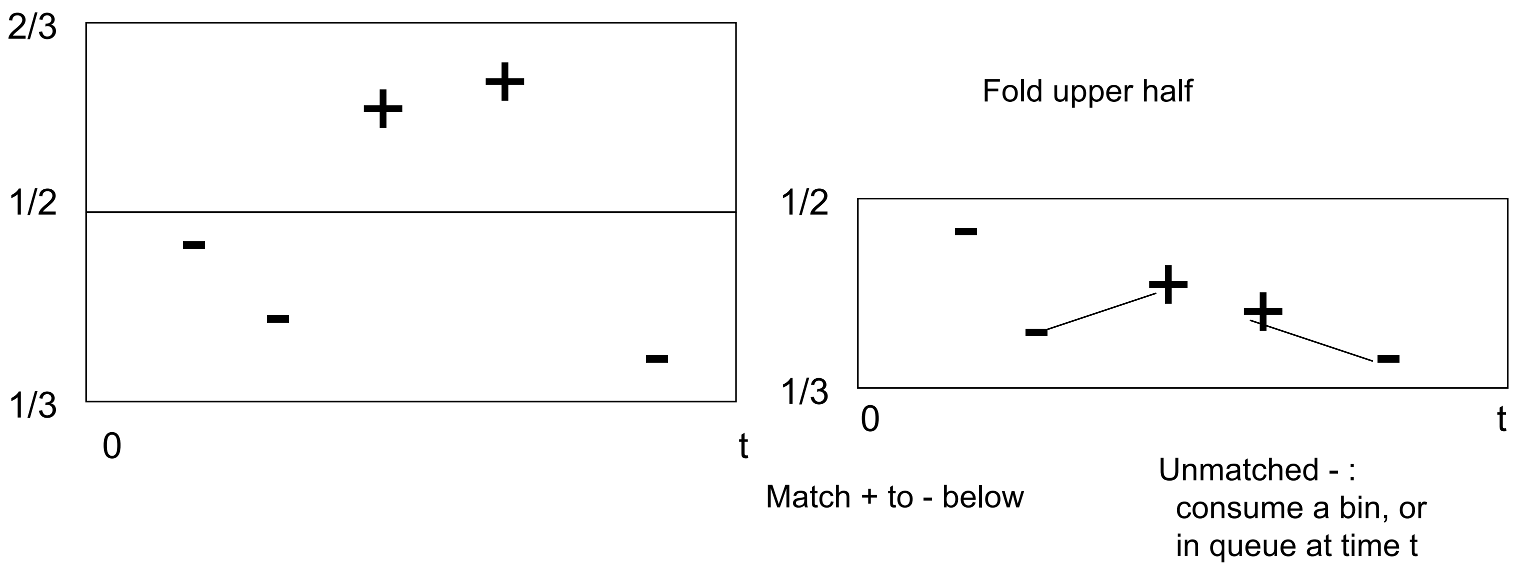
$$\mathbb{E}[D(0,t)] \le -c_1(1-\lambda)t + c_2\sqrt{t}\log^{3/4} t$$

$$\mathbb{E}\left[\left(\max_{s\in[0,t]} D(s,t)\right)^2\right] \le c_3 t \log^{3/2} t$$

$$\text{Choose } t \approx \frac{1}{(1-\lambda)^2}\log^{3/2}\frac{1}{1-\lambda}$$

Theorem: $\mathbb{E}[Q] \le c \cdot \frac{1}{1-\lambda}\log^{3/2}\frac{1}{1-\lambda}$

Lower Bound



- $\int_0^t Q(s)\,ds \;\geq\; \sum$(horizontal distances of matched)
- $Q(t) \geq \frac{1}{2}$(unmatched $-$) $+$ items in [1/2,1] $- t$
- $$\mathbb{E}[Q] \geq \mathbb{E}\left[\frac{1}{2t}\sum\text{(horizontal distances of matched)} + \frac{1}{2}\text{(unmatched } -\text{)}\right. \\ \left. -(1-\lambda)t\right]$$

Lower Bound

- $\mathbb{E}[Q] \geq \mathbb{E}\left[\frac{1}{2t}\sum(\text{horizontal distances of matched}) + \frac{1}{2}(\text{unmatched } -)\right] - (1-\lambda)t$

- Shor, building on Ajtai, Komlos & Tusnady: w.h.p.

$$\frac{1}{t}\sum(\text{horizontal distances of matched}) + (\text{unmatched } -) = \Omega(\sqrt{t \log t})$$

$$\mathbb{E}[Q] \geq c\sqrt{t \log t} - (1-\lambda)t$$

$$\text{Choose } t \approx \frac{1}{(1-\lambda)^2} \log \frac{1}{1-\lambda}$$

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

Conclusions

- Average $\longrightarrow$ Online $\longrightarrow$ Queueing
- Queueing variants have been little studied
- "Waste" of the static version is informative, but not the full story
- Other combinatorial problems with queueing counterparts?
- Applications?