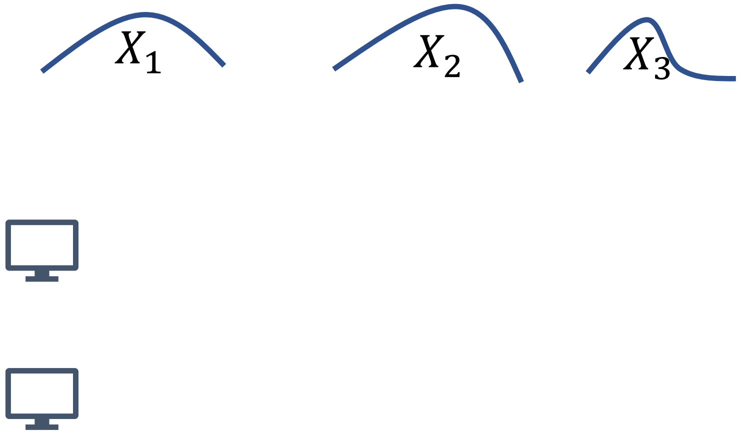
Stochastic Completion Time Minimization

Anupam Gupta, Benjamin Moseley, Rudy Zhou

Minimizing Completion Times for Stochastic Jobs via Batched Free Times
Symposium on Discrete Algorithms (SODA) 2023

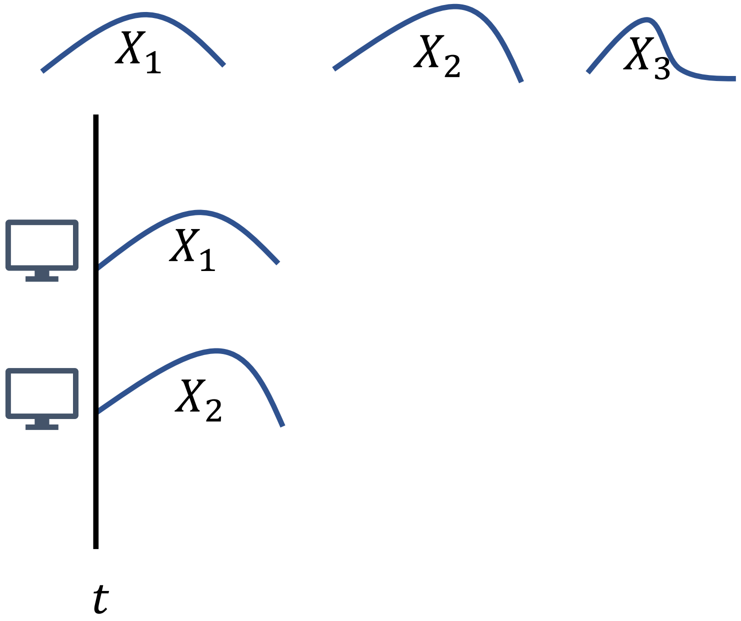
Minimizing Total Completion Time

- m identical machines
- n jobs with known, independent job-size distributions $X_j \sim$



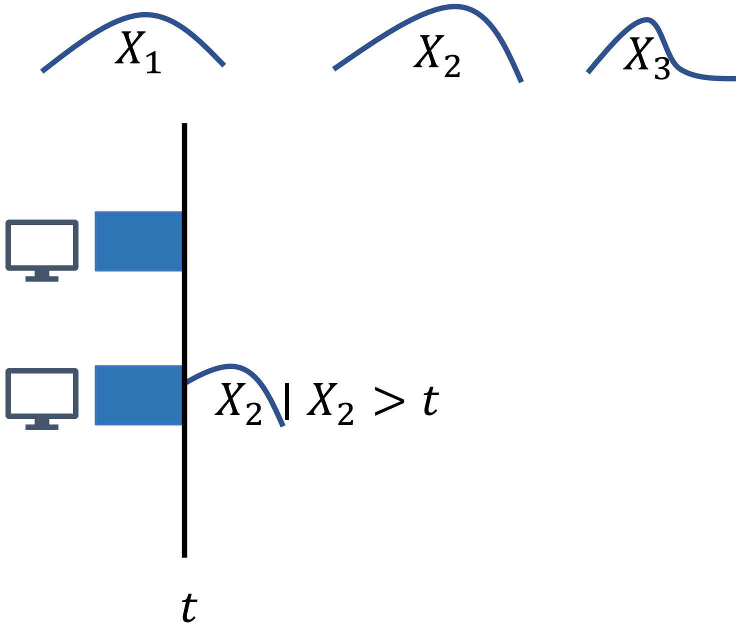
Minimizing Total Completion Time

- m identical machines
- n jobs with known, independent job-size distributions $X_j \sim$



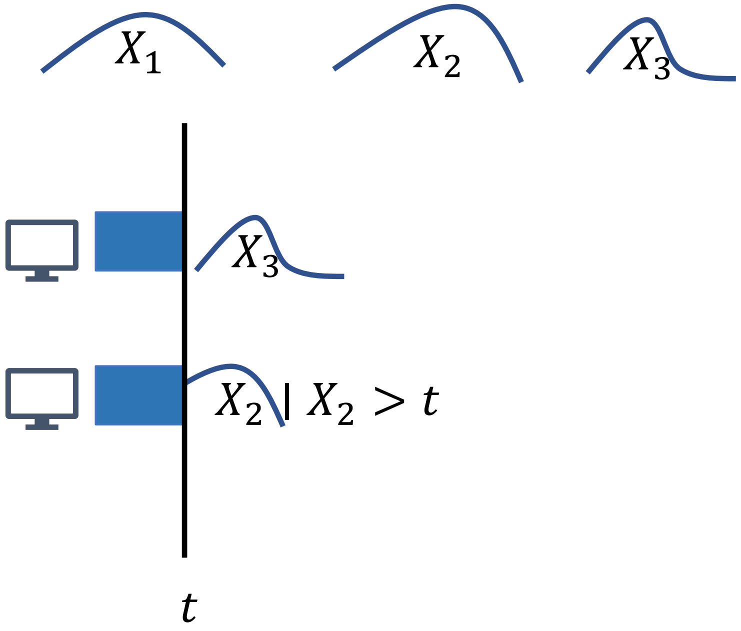
Minimizing Total Completion Time

- m identical machines
- n jobs with known, independent job-size distributions $X_j \sim$

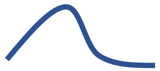


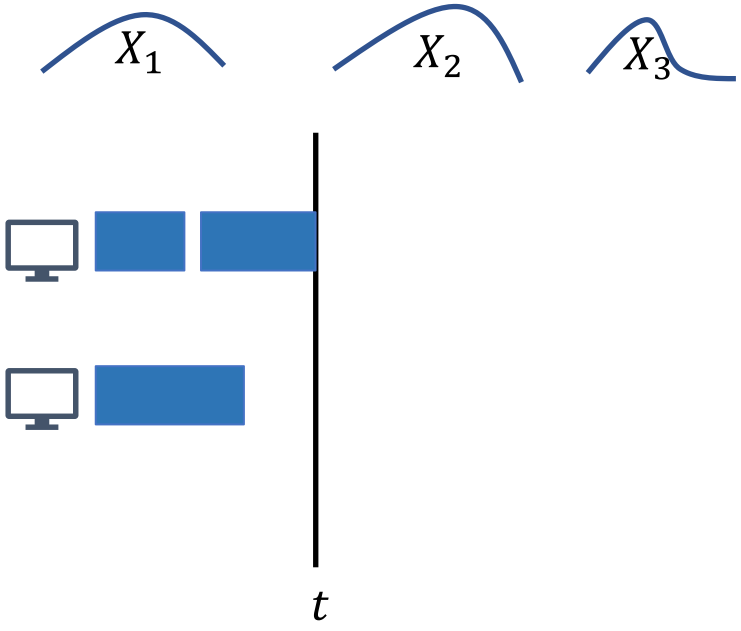
Minimizing Total Completion Time

- m identical machines
- n jobs with known, independent job-size distributions $X_j \sim$



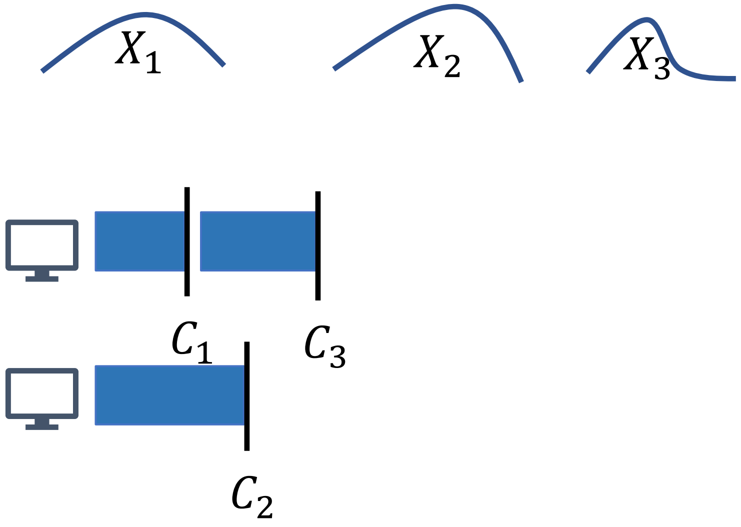
Minimizing Total Completion Time

- m identical machines
- n jobs with known, independent job-size distributions $X_j \sim$ 



Minimizing Total Completion Time

- m identical machines
- n jobs with known, independent job-size distributions $X_j \sim$



Objective: minimize expected total completion time, $\sum_j \mathbb{E}\, C_j$

…compared to optimal adaptive policy that also only knows distributions

Past Work

- Shortest Processing Time is optimal for deterministic jobs *[Bruno, Coffman Jr., Sethi, Commun. ACM 1974]*
- Stochastic jobs seem much harder
  - $O(\Delta)$-approximation, where $\Delta = \ max_j \ \frac{\mathbb{E}[X_j^2]}{\mathbb{E}[X_j]^2}$ is coefficient of variation via LP rounding *[Möhring, Schulz, Uetz, J. ACM 1999]*
  - …but all known LP's have integrality gap $\Omega(\Delta)$ *[Skutella, Sviridenko, Uetz, Math. Oper. Res. 2016]*
  - All distribution-independent approximations are $\Omega(m)$ [Im, Moseley, Pruhs, STACS 2015]
  - …and there are strong lower bounds for "greedy-like" policies *[Eberle, Fischer, Matuschke, Megow, Oper. Res. Lett. 2019]*

Our Results

Main Theorem: There exists an efficient algorithm that is $\tilde{O}(\sqrt{m})$-approximate for minimizing total completion time of stochastic jobs for Bernoulli jobs ($X_j \sim s_j \cdot Ber(p_j)$)

- First approximation that does not depend on coefficient of variation and is sublinear in number of machines

Our Results

Main Theorem: There exists an efficient algorithm that is $\tilde{O}(\sqrt{m})$-approximate for minimizing total completion time of stochastic jobs for Bernoulli jobs ($X_j \sim s_j \cdot Ber(p_j)$)

- First approximation that does not depend on coefficient of variation and is sublinear in number of machines

Idea: Optimize proxy objective (weighted free time)

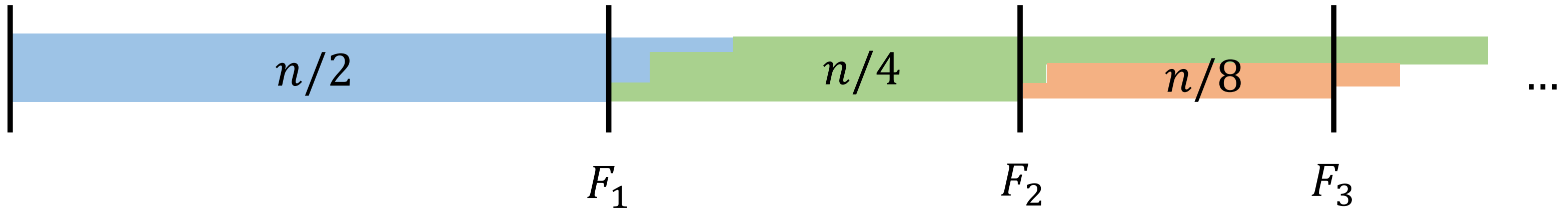
Weighted Free Time

- To minimize completion time, suffices to minimize starting time, $\sum_j \mathbb{E}\, S_j$ (shift of objective by $\sum_j \mathbb{E}\, X_j$)

Weighted Free Time

- To minimize completion time, suffices to minimize starting time, $\sum_j \mathbb{E}\, S_j$ (shift of objective by $\sum_j \mathbb{E}\, X_j$)

Free Time: Let F_k to be the earliest time a machine is free after starting $n - \frac{n}{2^k}$ jobs



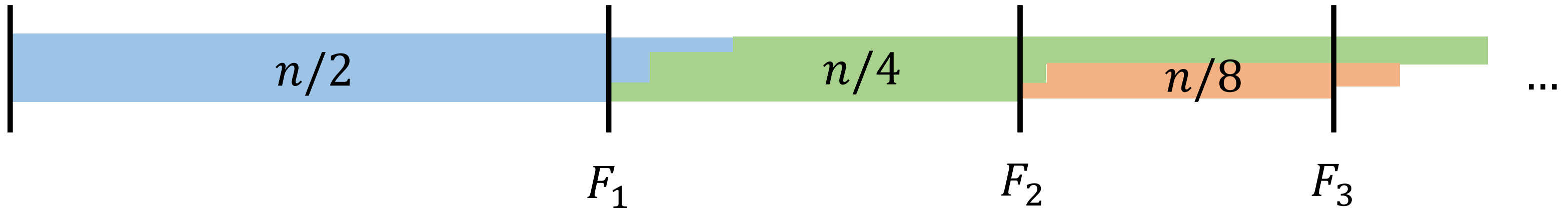
Weighted Free Time

- To minimize completion time, suffices to minimize starting time, $\sum_j \mathbb{E}\, S_j$ (shift of objective by $\sum_j \mathbb{E}\, X_j$)

Free Time: Let F_k to be the earliest time a machine is free after starting $n - \frac{n}{2^k}$ jobs

- To minimize starting time, suffices to minimize **weighted free time:** $\sum_{k=1}^{\log n} \frac{n}{2^k}\, \mathbb{E}\, F_k = \Theta(\sum_j \mathbb{E}\, S_j)$

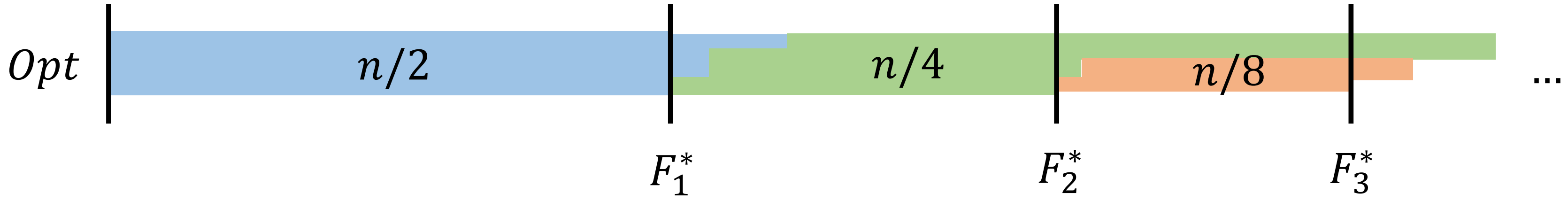
Proof Sketch: $\Theta(\frac{n}{2^k})$ jobs start in $[F_{k-1}, F_k]$



Minimizing Weighted Free Time

- Weighted free time: $\sum_{k=1}^{\log n} \frac{n}{2^k} \, \mathbb{E} \, F_k$

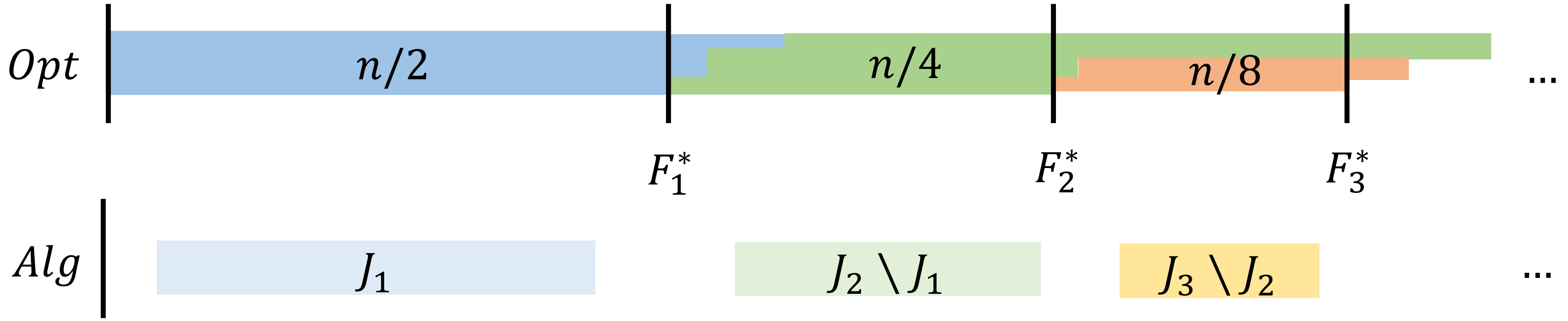
F_k is earliest time a machine is free after starting $n - \frac{n}{2^k}$ jobs



Minimizing Weighted Free Time

F_k is earliest time a machine is free after starting $n - \frac{n}{2^k}$ jobs

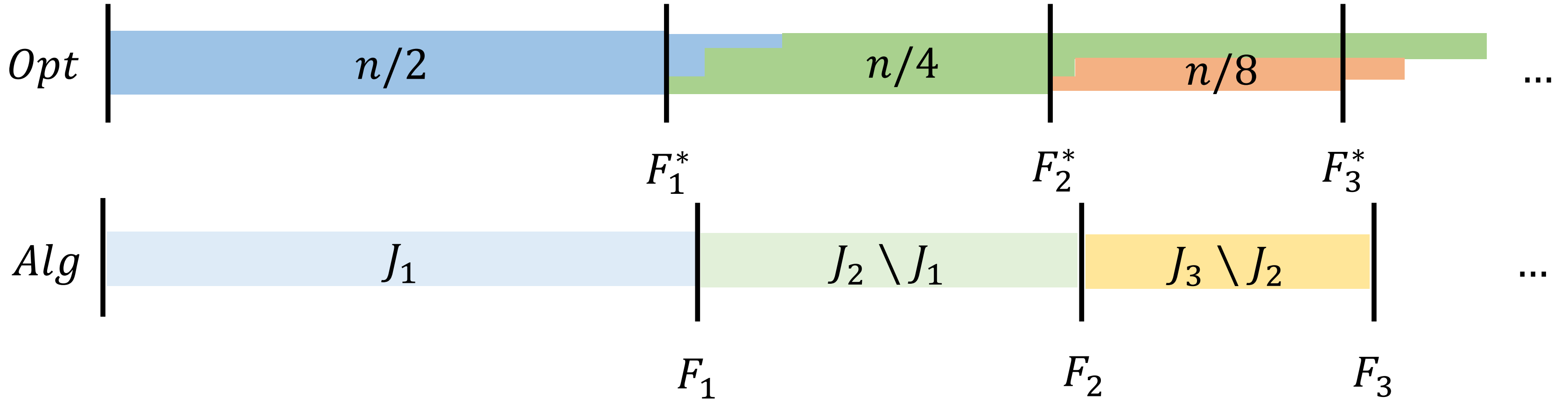
- Weighted free time: $\sum_{k=1}^{\log n} \frac{n}{2^k} \, \mathbb{E}\, F_k$
- **Subset Selection:** Choose nested sets of jobs $J_1 \subset J_2 \subset \cdots J_{\log n}$ such that J_1 comparable to Opt's first $n - n/2$, J_2 to first $n - n/4$, …



Minimizing Weighted Free Time

F_k is earliest time a machine is free after starting $n - \frac{n}{2^k}$ jobs

- Weighted free time: $\sum_{k=1}^{\log n} \frac{n}{2^k} \mathbb{E} F_k$
- **Subset Selection:** Choose nested sets of jobs $J_1 \subset J_2 \subset \cdots J_{\log n}$ such that J_1 comparable to Opt's first $n - n/2$, J_2 to first $n - n/4$, …
- **Batched Free Time Minimization:** Schedule $J_1 \subset J_2 \subset \cdots J_{\log n}$ subject to the batch constraint (must schedule J_k before $J_{k+1} \setminus J_k$) such that the free time of J_k is comparable to F_k^*



Algorithm

- Consider Bernoulli jobs $X_j \sim s_j \cdot Ber(p_j)$
- **Subset Selection:** To construct J_k: for each possible size parameter, exclude the $n/2^k$ jobs with largest probability parameters*

Algorithm

- Consider Bernoulli jobs $X_j \sim s_j \cdot Ber(p_j)$
- **Subset Selection:** To construct J_k: for each possible size parameter, exclude the $n/2^k$ jobs with largest probability parameters*
  - $|J_k| = n - \tilde{O}(\frac{n}{2^k})$
  - **For all k, we have $J_k \subset J_k^*$ for all realizations of job sizes, where J_k^* is the first $n - \frac{n}{2^k}$ jobs of Opt** **Proof Idea:** Exchange argument on optimal decision tree

* May assume there are $O(\log n)$-many distinct size parameters by standard discretization argument

Algorithm

- Consider Bernoulli jobs $X_j \sim s_j \cdot Ber(p_j)$
- **Subset Selection:** To construct J_k: for each possible size parameter, exclude the $n/2^k$ jobs with largest probability parameters*
  - $|J_k| = n - \tilde{O}(\frac{n}{2^k})$
  - **For all $\boldsymbol{k}$, we have $\boldsymbol{J_k \subset J_k^*}$ for all realizations of job sizes, where $\boldsymbol{J_k^*}$ is the first $\boldsymbol{n - \frac{n}{2^k}}$ jobs of $\boldsymbol{Opt}$** **Proof Idea:** Exchange argument on optimal decision tree
- **Batched Free Time Minimization:** Schedule each $J_k \setminus J_{k-1}$ in increasing order of size parameter

* May assume there are $O(\log n)$-many distinct size parameters by standard discretization argument

Algorithm

- Consider Bernoulli jobs $X_j \sim s_j \cdot Ber(p_j)$
- **Subset Selection:** To construct J_k: for each possible size parameter, exclude the $n/2^k$ jobs with largest probability parameters*
  - $|J_k| = n - \tilde{O}(\frac{n}{2^k})$
  - **For all k, we have $J_k \subset J_k^*$ for all realizations of job sizes, where J_k^* is the first $n - \frac{n}{2^k}$ jobs of Opt** **Proof Idea:** Exchange argument on optimal decision tree
- **Batched Free Time Minimization:** Schedule each $J_k \setminus J_{k-1}$ in increasing order of size parameter
  - **Within each $J_k \setminus J_{k-1}$ (other than the jobs with size zero), we schedule all jobs in increasing order of realized size**

* May assume there are $O(\log n)$-many distinct size parameters by standard discretization argument

Analysis Idea

Key Lemma: For all realizations of job sizes, we can write the weighted free time of Alg as:

$$\sum_{k=1}^{\log n} \frac{n}{2^k} F_k \approx O\left(\sum_{k=1}^{\log n} \frac{n}{2^k} \frac{Vol(J_k \setminus J_{k-1}(\leq F_k^*))}{m - |J_{k-1}(>F_k^*)|}\right)$$

Analysis Idea

Key Lemma: For all realizations of job sizes, we can write the weighted free time of Alg as:

Total size of all small (size $\leq F_k^*$) jobs in $J_k \setminus J_{k-1}$

$$\sum_{k=1}^{\log n} \frac{n}{2^k} F_k \approx O\left(\sum_{k=1}^{\log n} \frac{n}{2^k} \frac{Vol(J_k \setminus J_{k-1}(\leq F_k^*))}{m - \left|J_{k-1}(>F_k^*)\right|}\right)$$

Number of machines with no big (size $> F_k^*$) job before $J_k \setminus J_{k-1}$

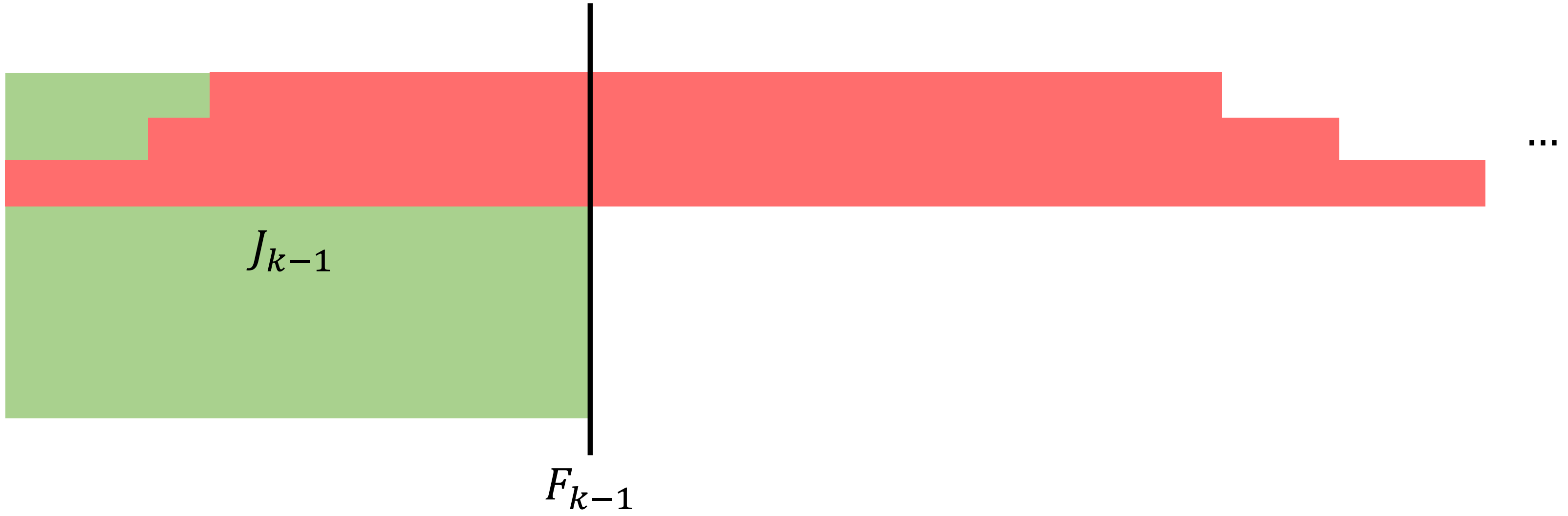
Analysis Idea

Key Lemma: For all realizations of job sizes, we can write the weighted free time of Alg as:

Total size of all small (size $\leq F_k^*$) jobs in $J_k \setminus J_{k-1}$

$$\sum_{k=1}^{\log n} \frac{n}{2^k} F_k \approx O\left(\sum_{k=1}^{\log n} \frac{n}{2^k} \frac{Vol(J_k \setminus J_{k-1}(\leq F_k^*))}{m - |J_{k-1}(>F_k^*)|}\right)$$

Number of machines with no big (size $> F_k^*$) job before $J_k \setminus J_{k-1}$



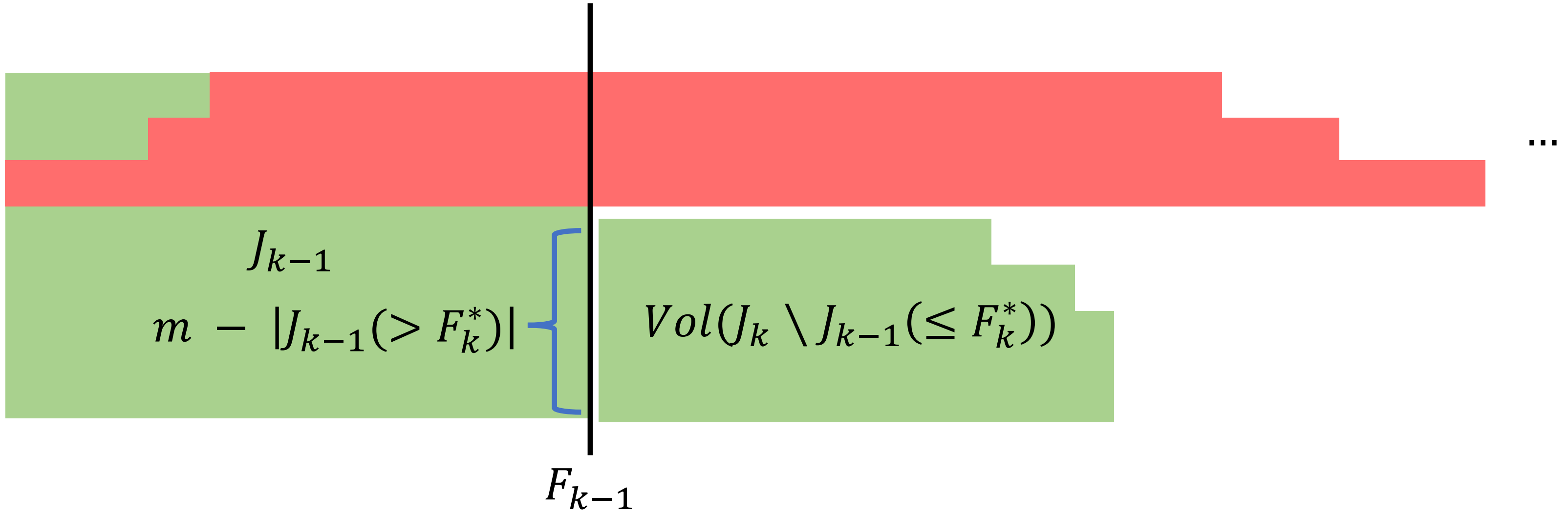
Analysis Idea

Key Lemma: For all realizations of job sizes, we can write the weighted free time of Alg as:

Total size of all small (size $\leq F_k^*$) jobs in $J_k \setminus J_{k-1}$

$$\sum_{k=1}^{\log n} \frac{n}{2^k} F_k \approx O\left(\sum_{k=1}^{\log n} \frac{n}{2^k} \frac{Vol(J_k \setminus J_{k-1}(\leq F_k^*))}{m - |J_{k-1}(> F_k^*)|}\right)$$

Number of machines with no big (size $> F_k^*$) job before $J_k \setminus J_{k-1}$



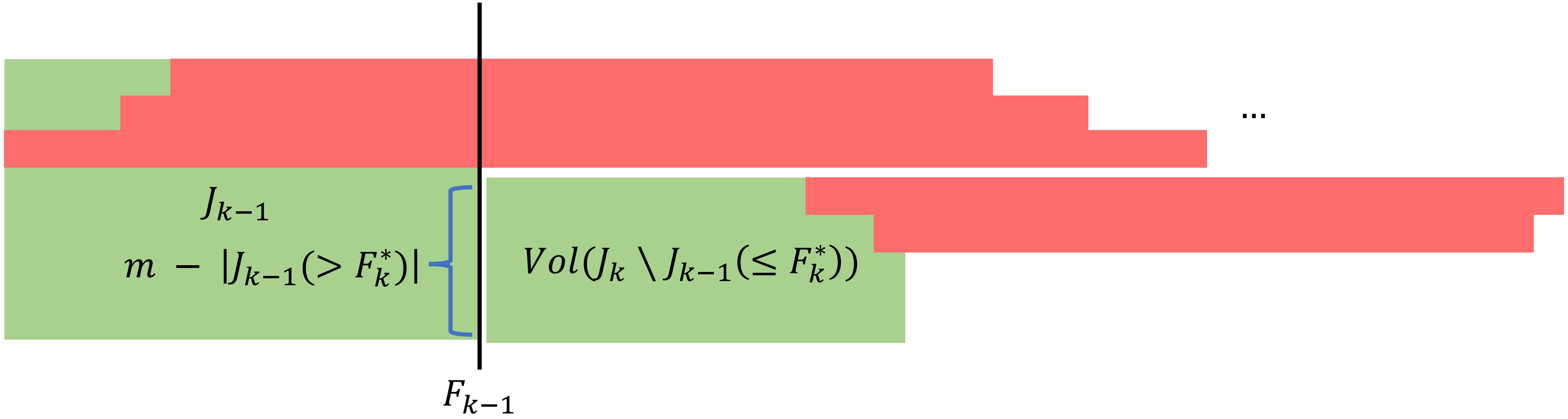
Analysis Idea

Key Lemma: For all realizations of job sizes, we can write the weighted free time of Alg as:

Total size of all small (size $\leq F_k^*$) jobs in $J_k \setminus J_{k-1}$

$$\sum_{k=1}^{\log n} \frac{n}{2^k} F_k \approx O\left(\sum_{k=1}^{\log n} \frac{n}{2^k} \frac{Vol(J_k \setminus J_{k-1}(\leq F_k^*))}{m - |J_{k-1}(> F_k^*)|}\right)$$

Number of machines with no big (size $> F_k^*$) job before $J_k \setminus J_{k-1}$



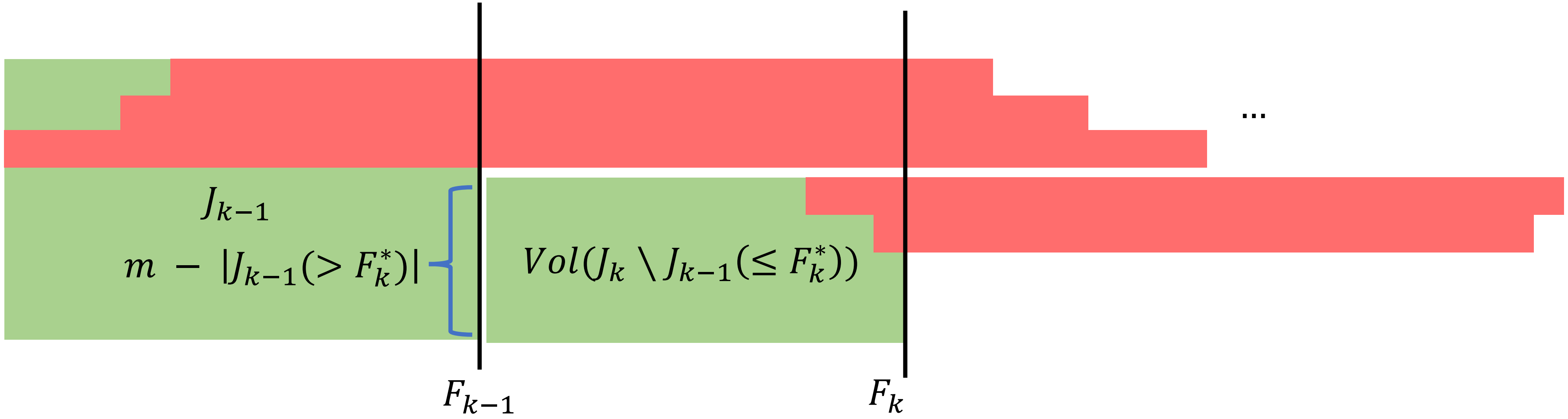
Analysis Idea

Key Lemma: For all realizations of job sizes, we can write the weighted free time of Alg as:

Total size of all small (size $\leq F_k^*$) jobs in $J_k \setminus J_{k-1}$

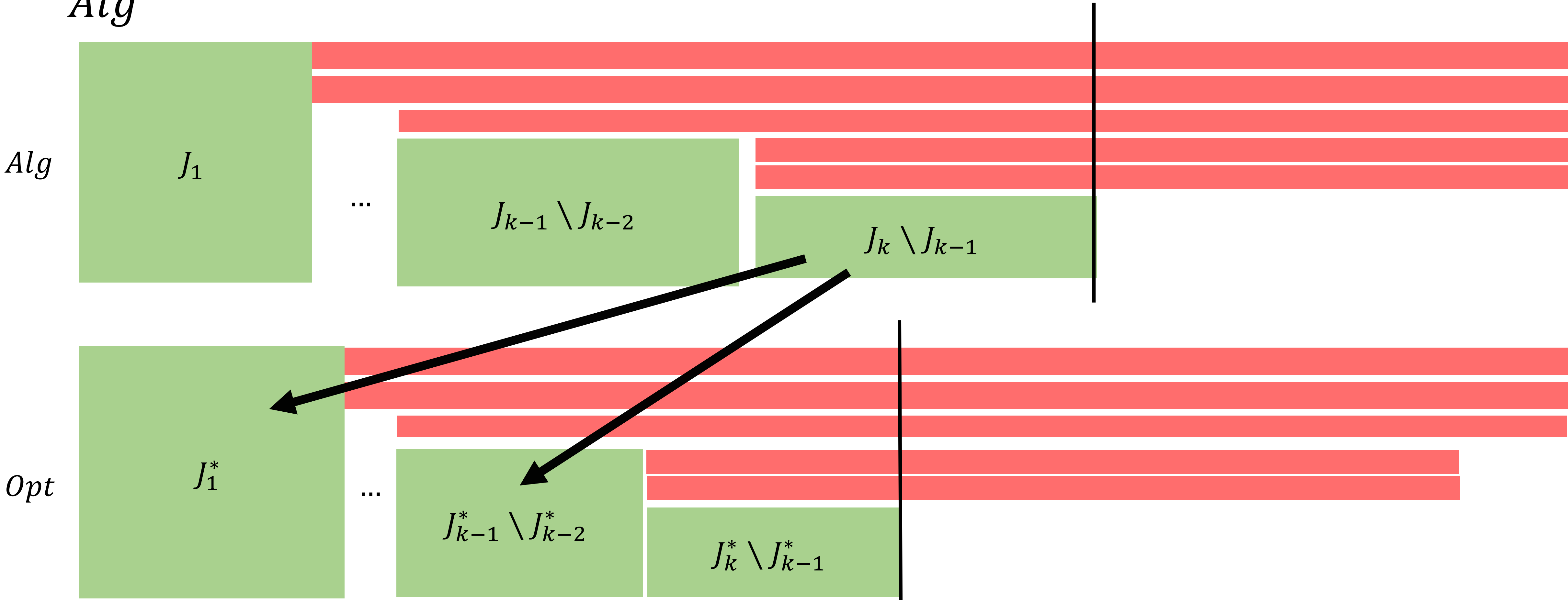
$$\sum_{k=1}^{\log n} \frac{n}{2^k} F_k \approx O\left(\sum_{k=1}^{\log n} \frac{n}{2^k} \frac{Vol(J_k \setminus J_{k-1}(\leq F_k^*))}{m - |J_{k-1}(> F_k^*)|}\right)$$

Number of machines with no big (size $> F_k^*$) job before $J_k \setminus J_{k-1}$



Analysis Idea

- **Problem:** Opt may schedule same volume in much earlier batches than Alg



Analysis Idea

- **Problem:** Opt may schedule same volume in much earlier batches than Alg

Lemma: In expectation, the number of machines with no big job decrease by at most a half between batches

$$\sum_{k=1}^{\log n} \frac{n}{2^k} F_k \approx O\left(\sum_{k=1}^{\log n} \frac{n}{2^k} \frac{Vol(J_k \backslash J_{k-1}(\leq F_k^*))}{m - |J_{k-1}(>F_k^*)|}\right)$$

Analysis Idea

- **Problem:** Opt may schedule same volume in much earlier batches than Alg

Lemma: In expectation, the number of machines with no big job decrease by at most a half between batches

$$\sum_{k=1}^{\log n} \frac{n}{2^k} F_k \approx O\left(\sum_{k=1}^{\log n} \frac{n}{2^k} \frac{Vol(J_k \setminus J_{k-1}(\leq F_k^*))}{m - |J_{k-1}(>F_k^*)|}\right)$$

- $\Rightarrow$ with high probability, the number of machines with no big job decrease by at most a half $\pm\, \tilde{\boldsymbol{O}}(\sqrt{\boldsymbol{m}})$ between batches
- **Solution:** If Opt decides to do a job in an earlier batch, the increase in number of available machines is offset by the increase in weight up to a $\tilde{O}(\sqrt{m})$-factor

Summary

- $\tilde{O}(\sqrt{m})$-approximation for minimizing total completion time of stochastic jobs for Bernoulli jobs ($X_j \sim s_j \cdot Ber(p_j)$)
- First approximation that is distribution-independent and $o(m)$ even for more restricted special cases
- **Idea:** Optimize proxy objective (weighted free time)