New Online Algorithms for Story Scheduling in Web Advertising

Susanne Albers
TU Munich

Achim Paßen
HU Berlin

Online advertising

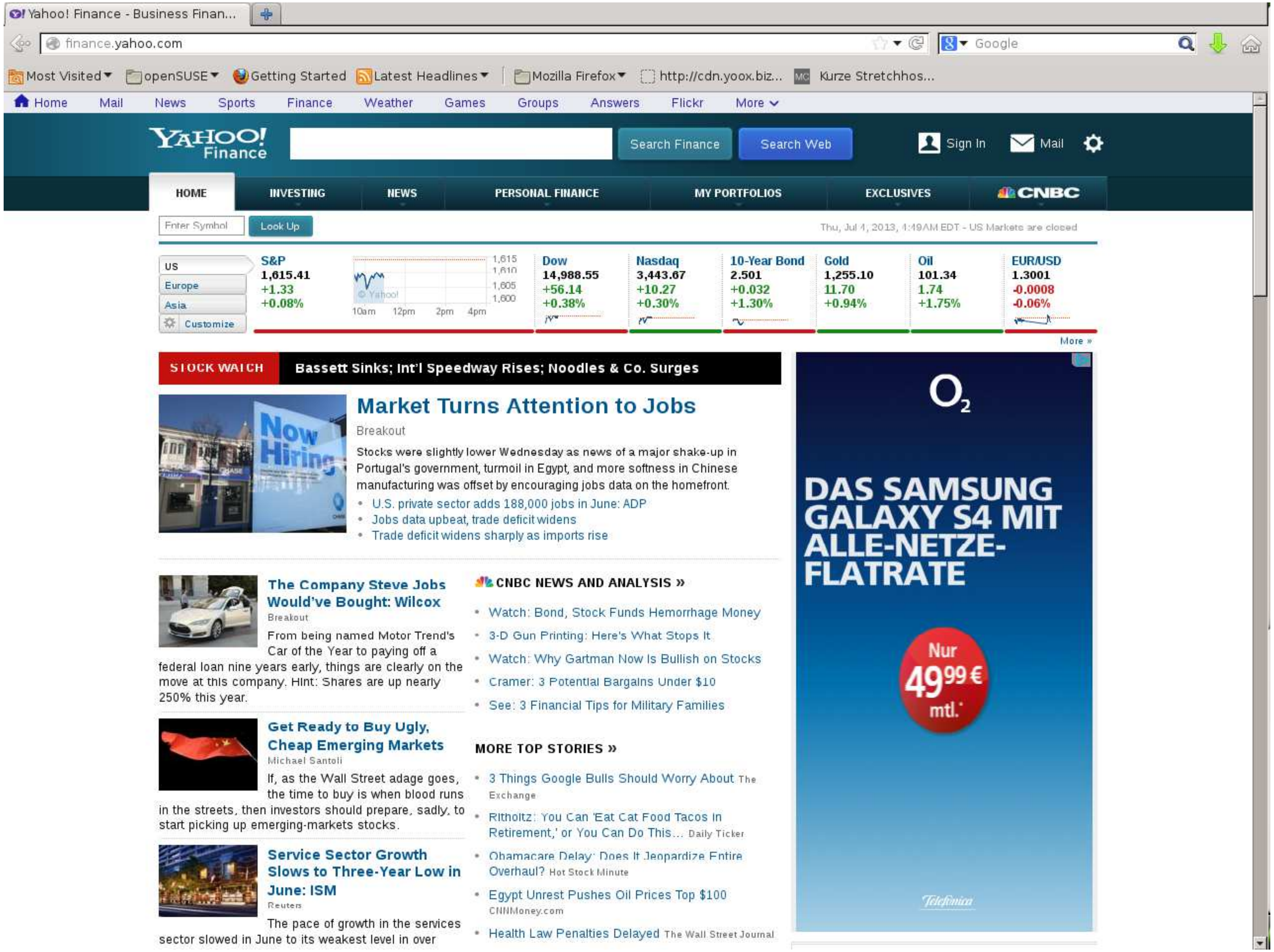
Worldwide online ad spending 2012/13: $ 100 billion

Expected to surpass print ad spending soon

Display advertising: images, videos, animations

Content shown depending on browsing history of user

Story boarding



Story boarding

- Maintain ad position of a web site during browsing session of a user.

- Position depicts image sequences of advertisers.
  Advertiser pay unit shown.

- Depending on history/state user becomes interesting for advertisers.

- Maximize revenue of session.

Model

- Session time is slotted
  Time t: user continues surfing with probability β $\quad 0 < \beta < 1$

- $I = J_1, \ldots, J_N$
  $J_i = (a_i, l_i, v_i)$
  a_i = arrival time $\qquad$ l_i = length $\qquad$ v_i = per-unit value

- Preemption: Job J_i may be scheduled for $< l_i$ time units

- $\max \sum_{t \geq 0} \beta^t v(t)$

  $v(t)$ = per-unit value of job scheduled at time t

Dasgupta, Ghosh, Nazerzadeh, Raghavan SODA'09

Competitive analysis

A: Online algorithm

$A(I)$

OPT: Offline algorithm

$OPT(I)$

A is c-competitive if for all I

$$A(I) \geq \frac{1}{c} \cdot OPT(I).$$

Previous results

- Lower bound: $c \geq 2$ for general β $\qquad c \geq \beta + \beta^2$

- Upper bound: $c = 7$

  ALG:

  $\forall t \;\; v_i$ of current $J_i \iff$ loss in delaying J_k with $v_k > v_i$ for 1 time unit

Dasgupta, Ghosh, Nazerzadeh, Raghavan SODA'09

Previous results

- Lower bound: $c \geq 2$ for general β $\qquad c \geq \beta + \beta^2$

- Upper bound: $c = 7$

- Jobs to be scheduled immediately upon arrival
  $c = \Omega(\sqrt{\log \mu / \log \log \mu})$ $\qquad \mu = \max\{l_{\max}/l_{\min},\ v_{\max}/v_{\min}\}$

- Total value of v_i gained only if entire J_i is shown
  $c = O(\log(v_{\max}/v_{\min}))$

Dasgupta, Ghosh, Nazerzadeh, Raghavan SODA'09

Our contribution

- Upper bound: $c = 4/(2-\beta)$

- Upper bound: $c = 1 + \Phi \approx 2.62$ $\Phi = \frac{1+\sqrt{5}}{2}$ Golden Ratio

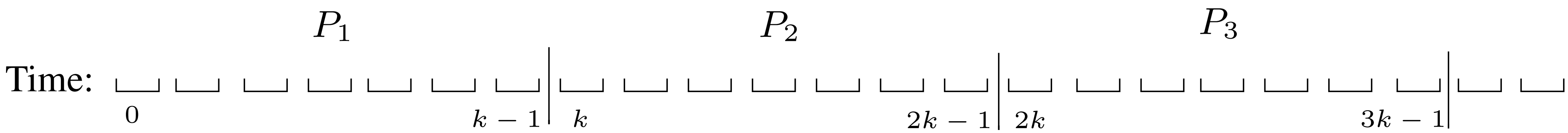
- Problem extension: Web page with m ad positions, where stories can be shown simultaneously; job migration not allowed

$$\max \sum_{t \geq 0} \sum_{j=1}^{m} \beta^t v(t,j)$$

$v(t,j)$ = per-unit value of job scheduled on ad position j at time t

Upper bound: $3.414 \approx 2/(2-\sqrt{2}) \leq c \leq 1/(3-2\sqrt{2}) \approx 5.828$

Algorithmic approach

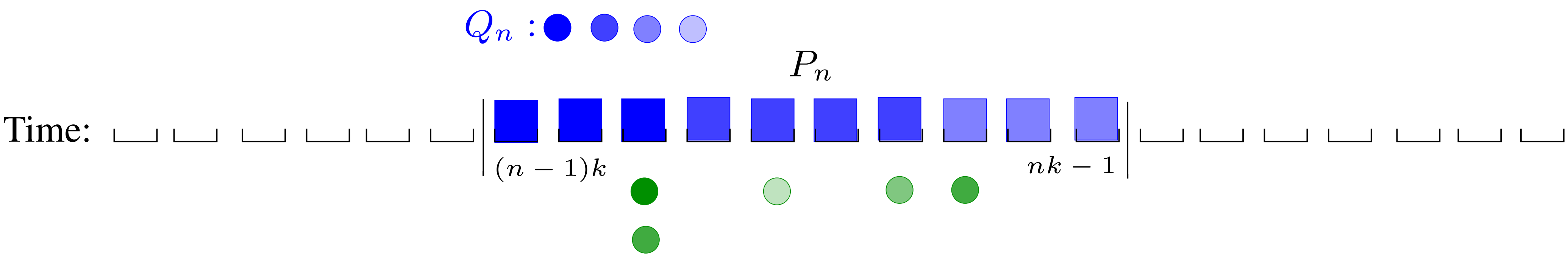


- Phases $P_1, P_2, P_3, \ldots$ of k consecutive time steps.

- Scheduling decisions are made at the beginning of the phase.

- Jobs arriving during the phase are ignored.

Simple algorithm



ALG1: Phase P_n: $Q_n = \{\text{unscheduled jobs } J_i \text{ with } a_i \leq (n-1)k\}$

Schedule jobs of Q_n in order of non-increasing per-unit value

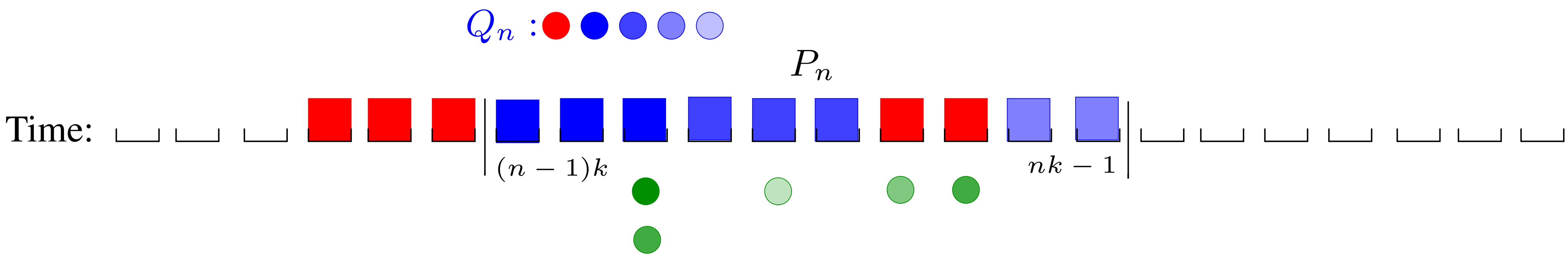
Preempt job at the end of P_n

Thm: $c = 1/(\beta^{k-1}(1-\beta^k))$

$c = 4/(2-\beta)$ for $k = \lceil -\log_\beta 2 \rceil$

$c = 1/(1-\beta)$ for $k = 1$

Refined algorithm



ALG2: Phase P_n: J_n = remainder of last job in P_{n-1}

$Q_n = \{J_n + \text{ unscheduled jobs } J_i \text{ with } a_i \leq (n-1)k\}$

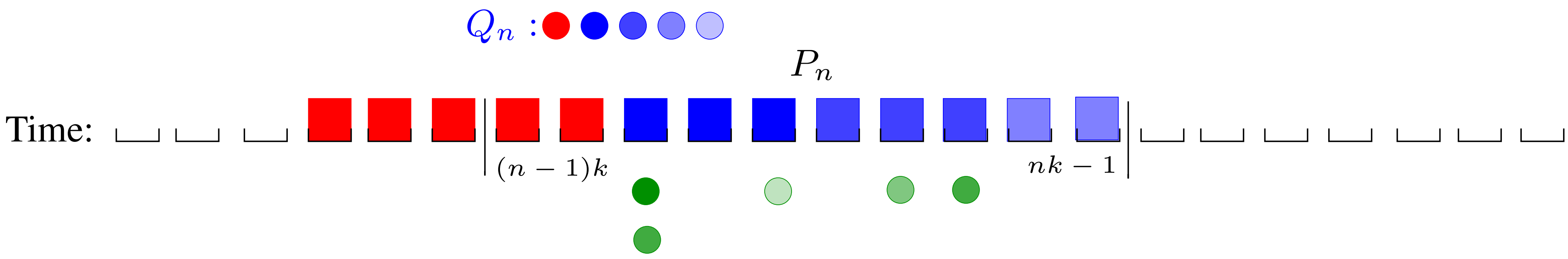
Schedule jobs of Q_n in order of non-increasing per-unit value

If J_n contained in schedule, move it to the front

Thm: $c = 1/\beta^{k-1} \cdot \max\{1/\beta^{k-1}, 1/(1-\beta^{2k}), \beta^{3k}/(1-\beta^k)\}$

$c = 1 + \Phi \approx 2.618$, where $\Phi = (1+\sqrt{5})/2$ for $k = \lfloor -\frac{1}{2}\log_\beta(1+\Phi)\rfloor + 1$

Refined algorithm



ALG2: Phase P_n: J_n = remainder of last job in P_{n-1}

$Q_n = \{J_n + \text{ unscheduled jobs } J_i \text{ with } a_i \leq (n-1)k\}$

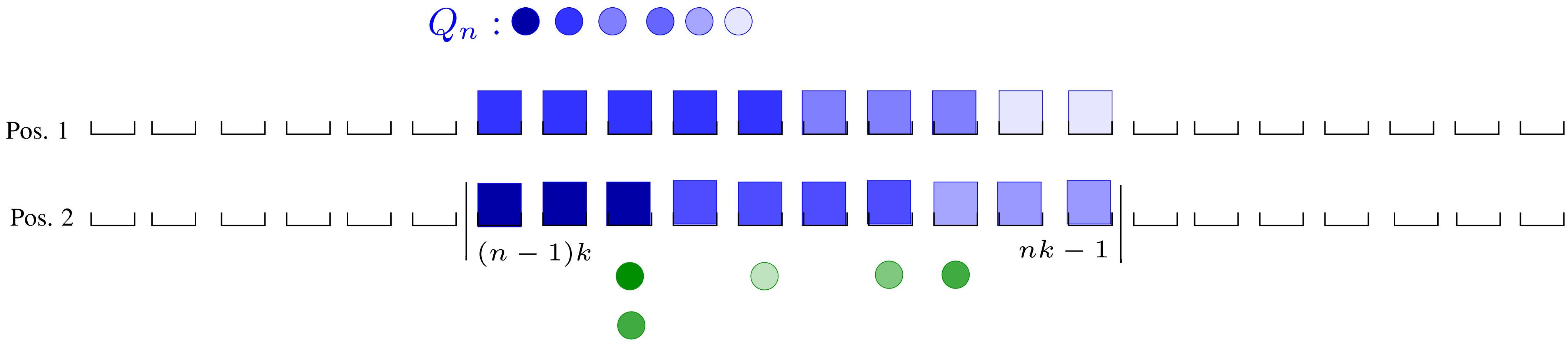
Schedule jobs of Q_n in order of non-increasing per-unit value

If J_n contained in schedule, move it to the front

Thm: $c = 1/\beta^{k-1} \cdot \max\{1/\beta^{k-1}, 1/(1-\beta^{2k}), \beta^{3k}/(1-\beta^k)\}$

$c = 1 + \Phi \approx 2.618$, where $\Phi = (1+\sqrt{5})/2$ for $k = \lfloor -\frac{1}{2}\log_\beta(1+\Phi)\rfloor + 1$

Algorithm for m ad positions



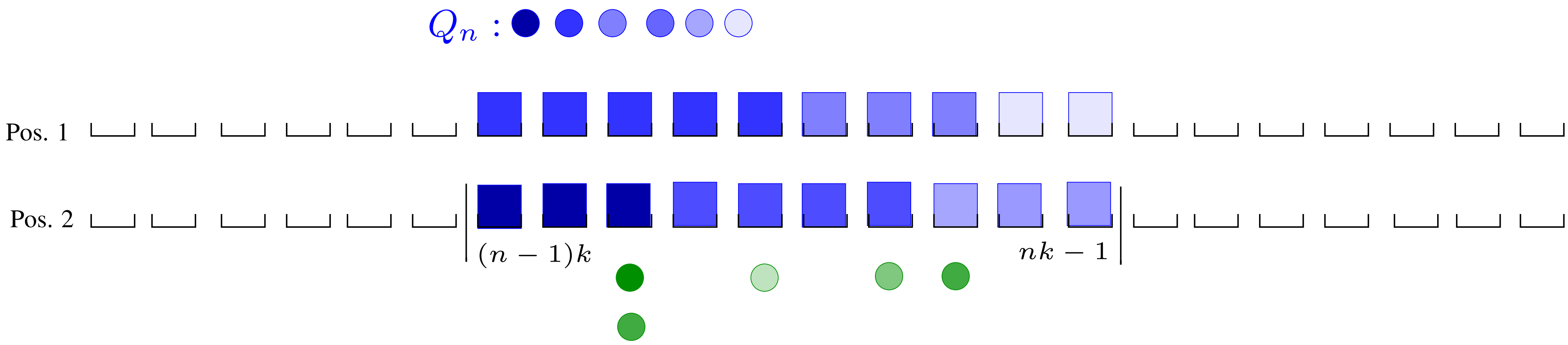
ALG3: Phase P_n: $Q_n = \{$unscheduled jobs J_i with $a_i \leq (n-1)k\}$

For $t = (n-1)k, \ldots, nk-1$, schedule m jobs of highest per-unit value

Preempt jobs at the end of P_n

Implementation: Units of a job are placed on same ad position

Algorithm for m ad positions



ALG3: Phase P_n: $Q_n = \{$unscheduled jobs J_i with $a_i \leq (n-1)k\}$

For $t = (n-1)k, \ldots, nk-1$, schedule m jobs of highest per-unit value

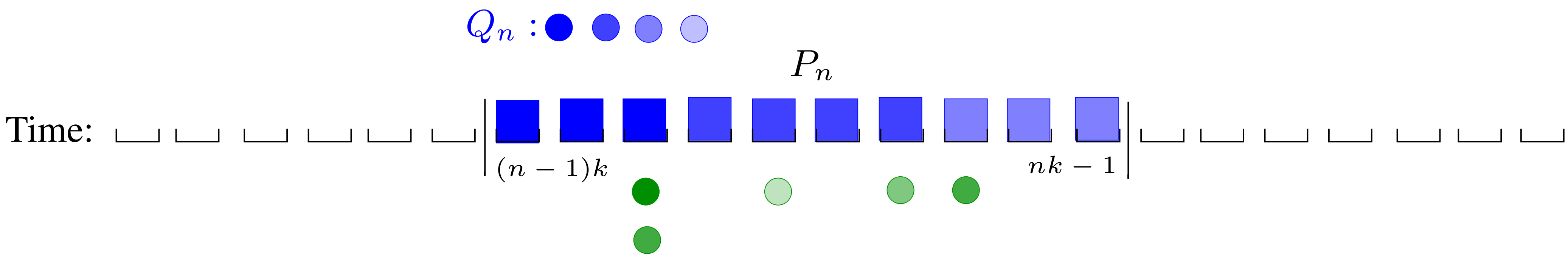
Preempt jobs at the end of P_n

Thm: $c = 1/\beta^{k-1} \cdot (1 + 1/(1-\beta^k))$

$c = (1 + 1/(1-\beta(2-\sqrt{2})))/(2-\sqrt{2})$ for $k = \lceil \log_\beta(2-\sqrt{2}) \rceil$

$3.414 \approx 2/(2-\sqrt{2}) \leq c \leq 1/(3-2\sqrt{2}) \approx 5.828$

Simple algorithm



ALG1: Phase P_n: $Q_n = \{$unscheduled jobs J_i with $a_i \leq (n-1)k\}$

Schedule jobs of Q_n in order of non-increasing per-unit value

Jobs of same value sorted in order of increasing arrival time

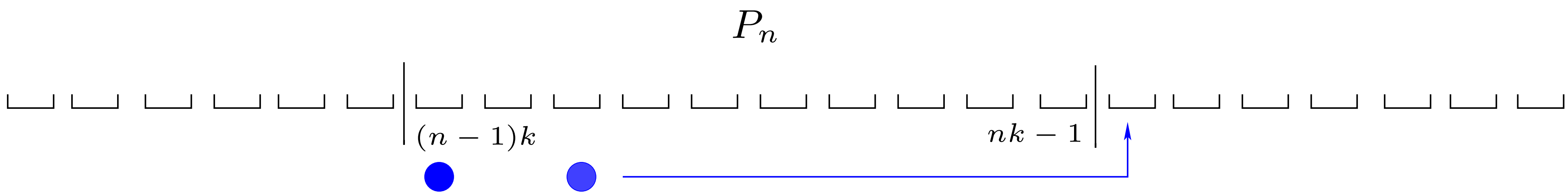
Preempt job at the end of P_n

Analysis simple algorithm

$I = J_1, \ldots, J_N$ $\qquad J_i = (a_i, l_i, v_i)$

k-quantized input

$I_k = J'_1, \ldots, J'_N$ $\qquad J'_i = (a'_i, l_i, v_i)$ $\qquad a'_i = k\lceil a_i/k \rceil$



Analysis simple algorithm

Lemma: $OPT(I_k) \geq \beta^{k-1} \cdot OPT(I)$

Proof: Shift optimal schedule for I by $k-1$ time units to the right.

Observation: $ALG1(I_k) = ALG1(I)$

Relaxed offline algorithm

$CHOP$: Optimal algorithm that may resume preempted jobs

Always schedule job with highest per-unit value.
Jobs with same per-unit value are scheduled
in the same order as in ALG1.

Observation: $CHOP(I_k) \geq OPT(I_k)$

Timing property

S = ALG1's schedule for I_k

S^* = $CHOP$'s schedule for I_k

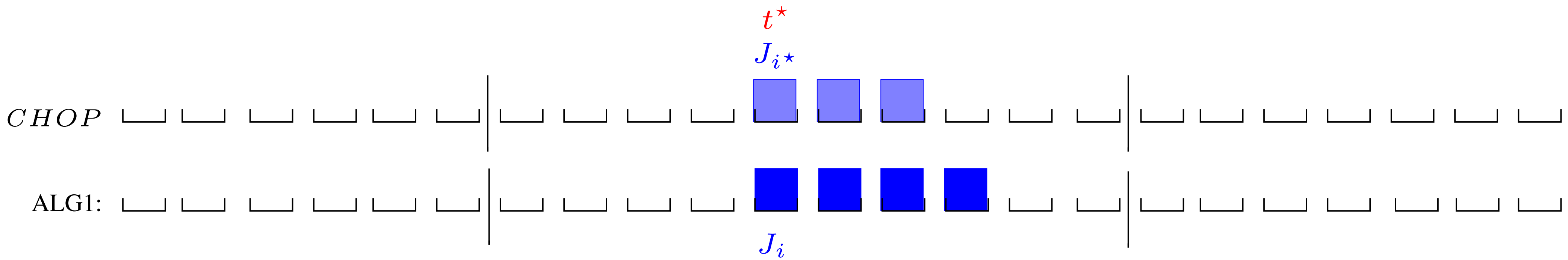
$t_S(i)$ = start time of J_i in S

$t_{S^*}(i)$ = start time of J_i in S^*

Lemma: $t_S(i) \leq t_{S^\star}(i)$

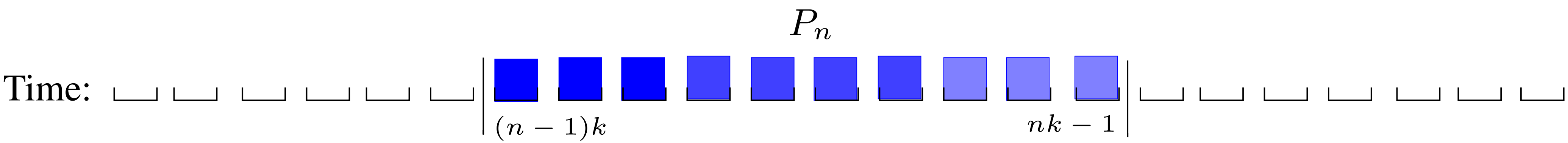
$t_S(i) \leq t_{S^\star}(i)$

Let $J_{i^\star}$ be first job in $S^\star$ with $t_S(i^\star) > t_{S^\star}(i^\star) = t^\star$



- $v_i \geq v_{i^\star}$
- At time $t^\star$ $CHOP$ has finished J_i
  $v_i > v_{i^\star}$: $\surd$
  $v_i = v_{i^\star}$: $CHOP$ schedules jobs of value v_i in the same order as ALG1
- $t_{S^\star}(i) \leq t^\star - l_i$ $\quad$ $t_S(i) \geq t^\star - l_i + 1$

Phase analysis



$\mathcal{J}_n$ = {jobs scheduled by ALG1 in P_n}

$\mathrm{ALG1}(P_n)$ = value achieved for $\mathcal{J}_n$ $\quad\quad$ $\mathrm{ALG1}(I_k) = \sum_n \mathrm{ALG1}(P_n)$

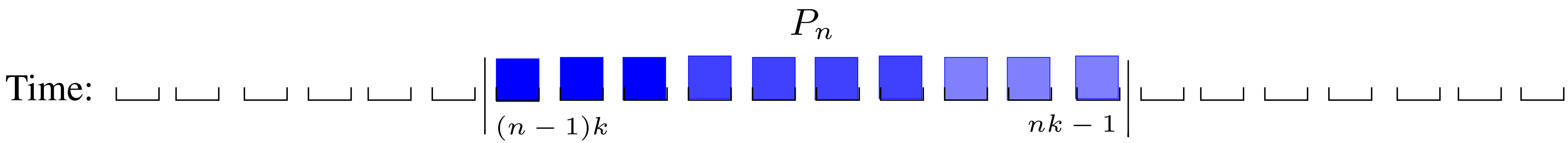
$CHOP(P_n)$ = value achieved for $\mathcal{J}_n$

Lemma: $CHOP(I_k) = \sum_n CHOP(P_n)$

Proof: For any J_i scheduled by $CHOP$, $\quad t_S(i) \leq t_{S^\star}(i)$

{jobs scheduled by $CHOP$} $\subseteq \bigcup_n \mathcal{J}_n$

Phase analysis



Lemma: $CHOP(P_n) \le \mathrm{ALG1}(P_n)/(1-\beta^k)$

Proof: J = last job of P_n with value v

$CHOP(P_n) \le \mathrm{ALG1}(P_n) + \text{extra value in scheduling preempted portion of } J$

$\mathrm{ALG1}(P_n)$ = $\sum_{t=(n-1)k}^{nk-1} \beta^t v = (\beta^{(n-1)k} - \beta^{nk})/(1-\beta) \cdot v$

extra value = $\sum_{t \ge nk} \beta^t v = \beta^{nk}/(1-\beta) \cdot v$

$$CHOP(P_n)/\mathrm{ALG1}(P_n) \le 1 + \beta^{nk}/(\beta^{(n-1)k} - \beta^{nk}) = 1/(1-\beta^k)$$

Wrapping up

$$CHOP(I_k) \leq \mathrm{ALG1}(I_k)/(1-\beta^k)$$

$$\beta^{k-1} \cdot OPT(I) \leq CHOP(I_k)$$

$$OPT(I) \leq \mathrm{ALG1}(I)/(\beta^{k-1}(1-\beta^k))$$

Analysis refined algorithm

- Take care of delays if last job of previous phase is continued.
- Loss of preempted jobs have to amortized over several phases; segments of up to three phases.

Analysis m ad positions

- $\exists\ S^\star$ value is at least as high as that of optimal schedule

- $S^\star$ schedules up to $2m$ jobs at any time

- $\forall t$: each J_i scheduled in $S^\star$ but not in S can be mapped
  (a) to a unit $v_{i'} \geq v_i$ scheduled in S or
  (b) to a job preempted by S at time $t' < t$

Open problems

- Tight bounds for deterministic algorithms

- Design randomized algorithms