02.02.23

- Algorithm PLTR:

  - initialise $m_t \leftarrow m$

    $l_t \leftarrow 0$ for all $t \in [0, d_{max}]$

  - for $k = m$ to 1

    ~~while t~~

    - $t = 0$
    - while $t < d_{max}$
      - $t = \text{keepidle}(k, t)$
      - $t = \text{keepactive}(k, t)$

- keepidle(k, t):

  - search for maximal $t' \geq t$ s.t. # binary search

    $\exists$ feasible schedule with $m'_{t''} = k - 1 \quad \forall t'' \in [t, t')$
  - set $m_{t''} := k - 1$ for all these t''
  - return t'

- keepactive(k, t):

  - search for maximal $t' \geq t$ s.t.

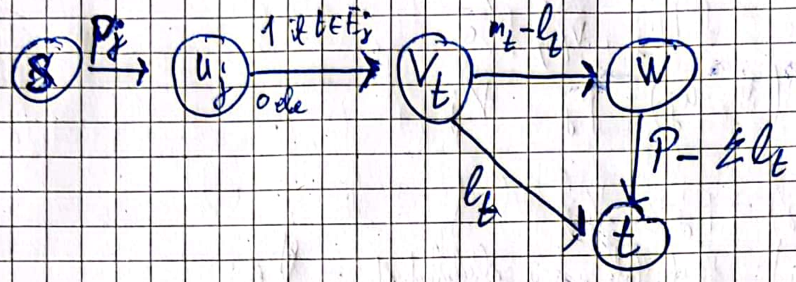
    $\exists$ feasible schedule with

    $l'_{t''} = \max\{k, l_{t''}\} \quad \forall t'' \in [t, t']$

  - set $l^*_{t''} :=$ "

  - return t'

- Flow calculation for [illegible] / Beispiele
  (i.e. with l_t, m_t)



- Exists feasible schedule respecting l_t, m_t $\Leftrightarrow$ max flow $= P$

$\Leftarrow$:

- Let f be a s-t flow of $v(f) = P \in \mathbb{N}_0$
- We construct a feasible schedule respecting l_t, m_t from f
  - $\forall j \in J$: if $f(u_j, V_t) = 1$, then schedule j at t
  - Since $v(f) = P$ and $c(s, V \setminus \{s\}) = P$:
    - $f_{in}(u_j) = p_j \quad \forall j$
    - $\Rightarrow f_{out}(u_j) = \sum_{t \in E_j} f(V_t) = p_j$
  - $\Rightarrow$ feasible schedule: every j is scheduled in p_j distinct time slots contained in E_j
  - Respects l_t, m_t:
    - $c(V_t, w) + c(V_t, t) \leq m_t - l_t + l_t$
      flow conserv.
      $\Rightarrow \forall t$: at most m_t jobs scheduled at t
  - Respects l_t:
    - ~~$c(V_t, t)$ $\delta(\{s\} \cup \{u_j\} \cup \{V_t\} \cup \{w\})$~~
    - $c(V \setminus \{t\}, \{t\}) = P$
    - $\Rightarrow f(V_t, t) = l_t \quad \forall t$
    - $\Rightarrow$ by flow conservation: $f_{in}(V_t) \geq l_t \quad \forall t$
    - $\Rightarrow \forall t$: $\geq l_t$ jobs scheduled at t

"$\Rightarrow$": ~~Let~~ Consider a schedule respecting l_t, m_t

- We build a flow of value P:
- If j is scheduled at t (hence $t \in E_j$), define $f(u_j, v_t) = 1$
  else $f(u_j, v_t) = 0$
- Define $f(s, u_j) = p_j \quad \forall j$

$\Rightarrow f_{in}(u_j) = p_j \quad \forall j$

$f_{out}(u_j) = \#$ of slots, j is scheduled at $\overset{fea}{=} p_j$

- define $f(v_t, t) = l_t \quad \forall t$
- define $f(v_t, w) = f_{in}(v_t) - l_t$
  - $f(v_t, w) \leq m_t - l_t$:
    - $f_{in}(v_t) = \#$ jobs scheduled at $t \leq m_t$
  - ~~$f_{in}(v_t) - f_{out}(v_t)$~~
  - $f_{out}(v_t) = f_{in}(v_t) - l_t + l_t = f_{in}(v_t)$

- define $f(w, t) = P - \sum l_t$
  - $f_{in}(w) = \sum_t f_{in}(v_t) - l_t$

$$= \sum_t \# \text{jobs scheduled at } t - \sum_t l_t$$

$$\overset{\text{schedule feasible}}{=} P - \sum_t l_t = f_{out}(w)$$

Mit CamScanner gescannt

Approximation Guarantee

- Preliminaries:
  - $v_S[j,Q]$: # of slots of Q, j is scheduled at in S
  - $fv[j,Q] = \max\{0; p_j - |E_j \setminus Q|\}$
    - # of slots, j has to be scheduled in Q in every feasible schedule
  - $uv_S[j,Q] = v_S[j,Q] - fv[j,Q]$
  - $pv[j,Q] = \min\{|E_j \cap Q|, p_j\}$
    - $\# \; pv[j,Q] \geq v[j,Q] \geq fv[j,Q]$
  - $V[J',Q] := \sum_{j \in J'} v[j,Q]$
  - $s_S[j,Q]$ = # of slots in $Q \cap E_j$ which j is not scheduled in for S
  - $\phi(Q) = \frac{fv[J,Q]}{|Q|}$
  - $\hat{\phi}(Q) = \max_{Q' \subseteq Q} \phi(Q')$
  - $V[Q] := V[J,Q]$

Lemma: feasibility characterisation with def, excess (with l_t, m_t)

- Definition: $\cdot\ def(Q) := pr[Q] - \sum_{t\in Q} m_t$

  $\cdot\ exc(Q) := \sum_{t\in Q} l_t - pr[Q]$

- Lemma: an instance with l_t, m_t is feasible

  $\Leftrightarrow \forall Q:\ def(Q) \leq 0$ and $exc(Q) \leq 0$

- "$\Rightarrow$" trivial

  if $exc(Q) > 0$ for some Q, some l_t for $t \in Q$ cannot be met

  if $def(Q) > 0$ " m_t "

"$\Leftarrow$": Consider an infeasible instance (with l_t, m_t):

~~We show that def'~~

$\Rightarrow$ corresponding $flow_{max}$ f has value $v(f) < P$

$\Rightarrow \exists$ min-cut $(S, \bar{S})$ s.t. $c(S) < P$

- Lemma: $\forall$ cut $(S,\bar{S})$: $c(S) \geq P - def(Q(S))$

  or $c(S) \geq P - excess(Q(\bar{S}))$

  where $Q(S) := \{ t \mid v_t \in S \}$

- ~~Case I: Let S be a cut $(S,\bar{S})$.~~

~~Case I~~

Let $(S, \bar{S})$ be a cut, let $\mathcal{J}(S) := \{j \mid u_j \in S\}$

Case I: $w \notin S$

Node of S	contrib. to $c(S)$
s	$\sum_{j \in \mathcal{J}(\bar{S})} p_j$
u_j	number of $v_t \in \bar{S}$ s.t. $t \in E_j$ $= \lvert E_j \setminus Q(S) \rvert$ $\overset{\circledast}{\geq} p_j - fo[j, Q(S)]$
v_t	$l_t + m_t - l_t = m_t$

$\circledast$ $fo[j, Q(S)] = \{\max\{0, \underbrace{p_j - \lvert E_j \setminus Q(S) \rvert}_{\geq}\}\}$

$$\Rightarrow c(S) \geq \sum_{j \in \mathcal{J}(\bar{S})} p_j + \sum_{j \in \mathcal{J}(S)} p_j - fo[j, Q(S)] + \sum_{t \in Q(S)} m_t$$

$$= P - fo[\mathcal{J}(S), Q(S)] + \sum_{t \in Q(S)} m_t$$

$$\geq P - def(Q(S))$$

Case II: $w \in S$

Node of S	contrib. to $c(S)$

Case II $w \in S$

Node of S	contrib to $c(S)$
s	$\sum_{j \in \partial(\bar{S})} p_j \geq \sum_{j \in \partial(\bar{S})} pr[j, Q(\bar{S})]$
u_j	$\lvert E_j \setminus Q(S) \rvert = \lvert E_j \cap Q(\bar{S}) \rvert \geq \min\{p_j, E_j \cap Q(\bar{S})\} = pr[j, Q(\bar{S})]$
v_t	l_t
w	$P - \sum_t l_t$

$$\Rightarrow c(S) \geq P - \sum_{t \in Q(\bar{S})} l_t + pr[Q(\bar{S})]$$
$$= P - \text{exc}(Q(\bar{S}))$$

Since $c(S) < P$, we have $\text{def}(Q) > 0$ or ~~def~~ $\text{excess}(Q) > 0$ for some Q

Mit CamScanner gescannt

03.02.23

Lemma „Critical sets" (of timeslots)

- For every activation slot t of some proc. k in S_{PLTR}:
  - $\exists Q \subseteq [0, d_{max}]$ s.t.
    - $t \in Q$
    - $fv[Q] = v[Q]$ (1)
    - $\forall t' \in Q: v[t'] \geq k-1$ (2)
    - $\forall t' \in Q$ with $t' \geq t$: $v[t'] \geq k$ (3)
- Proof: Suppose for c.d. there is some activation t of proc. k s.t. no such Q exists

  $\Rightarrow \forall Q$ s.t. $t \in Q$ and (2), (3):

  $$v[Q] > fv[Q]$$

- We show that PLTR would have extended the idle interval on proc. k which ends before t
- Consider the step in PLTR when t was the result of Keepidle on proc. k and the corresponding lower and upper bounds $m_{t'}, l_{t'}$ for $t' \in [0, d_{max}]$ after the calculation of t by Keepidle
- Make the following modification: decrease m_t by 1
- Note that at this point $m_{t'} \geq k$ $\forall t' > t$ and

  $$m_{t'} \geq k-1 \quad \forall t'$$

- Consider Q s.t. $t \in Q$, ~~(2), (3) and hence~~ $\neg$(1)

  before: $m_Q := \sum_{t' \in Q} m_{t'} \overset{(*)}{\geq} v[Q] > fv[Q]$

  $\Rightarrow$ after: $m_Q \geq fv[Q]$

  (*) since upper bounds monotonically decrease in PLTR

- Consider Q s.t. $t \in Q$, $\neg$(2):

  $\Rightarrow \exists t' \in Q$ s.t. $v[t'] < k-1 \Rightarrow m_{t'} \geq k-1 > v[t']$

  $\Rightarrow$ before: $m_Q >$ ~~...~~ $v[Q] \geq fv[Q]$

  after: $m_Q \geq fv[Q]$

03.02.23

- Consider Q s.t. $t \in Q$, $\neg$ ③:
  - $\Rightarrow \exists t' \in Q$ s.t. $v[t'] < k$ and $t' > t$
  - $\Rightarrow m_{t'} \geq k > v[t']$
  - $\Rightarrow$ before: $m_Q > v[Q] \geq fr[Q]$
  - after: $m_Q \geq fr[Q]$

$\Rightarrow t$ cannot be the result of Deepidle at this step in PLTR ↯

$\Rightarrow \exists Q$ s.t. $t \in Q$, ①,②,③

- We call such Q for activations t on proc. k: „tight set Q_t over "
- Critical set C_t over "
  is maximum of $\{Q_t\}$ in regard to ϕ, i.e.
  - $C_t := \arg\max \{ \phi(Q) \mid Q \subseteq [0, d_{max}]$ with $t \in Q$, ①,②,③
  - As the set of C_t's is clearly closed under union, we take C_t to be the inclusion maximal critical set over activation t for the sake of uniqueness

- Definitions based on C_t
  - total order $\precsim$ on $\{C_t\}$ based on t's:

    from top left to bottom right

    (same order in which plt proceeds)
  - rank : $\{C_t\} \to \mathbb{N}$

    Mapping to $\mathbb{N}$ corresponding to $\precsim$, i.e.

    $$\text{rank}(C_t) \geq \text{rank}(C_{t'}) \Leftrightarrow C_t \succsim C_{t'}$$

~~# drop subscripts, just C now~~

  - $\text{crit}(C_t) = C \Leftrightarrow t$ is activation of proc. C

    (C is the highest proc. activated at t)
  - Extension to general time slots, intervals
    - $\text{rank}(t) := \begin{cases} \max\{\text{rank}(C) \mid t \in C\} & t \in C \text{ for some } C \\ 0 & \text{else} \end{cases}$
    - $\text{crit}(t) := \begin{cases} \max\{\text{crit}(C) \mid t \in C\} & \text{"} \\ 0 & \text{else} \end{cases}$

    - Interval D:
      - $\text{rank}(D) = \max\{\text{rank}(t) \mid t \in D\}$
      - $\text{crit}(D) =$ " crit "
  - An interval V is called a valley on rank(C) if
    - $C \sim V$
    - V is maximal (in regard to "$\supseteq$")
  - Jobs:

    rank$(j) = \max\limits_{t:\, j \text{ is scheduled at } t} \text{rank}(t)$

    crit$(j) =$ " crit(t)

Mit CamScanner gescannt

~~Let C be a cri~~

- Valleys V_ℓ, V_r:
- Let C be critical set
- An interval D is said to "span" C if
  - $D \cap C$ contains only (full) ~~and~~ subintervals of C and at least one subinterval
- For critical set C, interval D spanning C we define:
  - V_ℓ: valley of maximal rank ending at $\min_{C,} D - 1$
  - V_r: valley " beginning at $\max C \cap D + 1$
- Lemma Valley:

For every C (of $\mathrm{crit}(C) =: c$), every interval D spanning C:

$$\text{if } \phi(C \cap D) \leq c - \delta \quad \text{for some } \delta \in \mathbb{N}$$

$$\text{then } |J_{V_\ell}| + |J_{V_r}| \geq \delta \quad \text{and } V_r \text{ or } V_\ell \text{ exist}$$

where J_V ~~is the~~ $\subseteq J$ is the set of jobs scheduled at every $t \in V$ (or ϕ if V is empty)

- Proof: By choice of C: ~~$\forall [\delta] \geq \delta$~~ $v[C \cap D] \geq \cancel{A}(c-1) \cdot |C \cap D|$

Case I: $v[C \cap D] > (c-1) \cdot |C \cap D|$

$$\Rightarrow \text{if } \frac{\mathrm{fv}[C \cap D]}{|C \cap D|} \leq c - \delta \text{ then } \frac{\mathrm{uv}[C \cap D]}{|C \cap D|} > \delta - 1$$

$\Rightarrow$ there are at least δ jobs j scheduled in $C \cap D$ with ~~that~~ $\mathrm{uv}[j, C \cap D] > 0$.

Such jobs must have $E_j \cap (C \setminus D) \neq \phi$ and hence must be contained in J_{V_ℓ} or J_{V_r} and V_ℓ or V_r must exist

Case II: $v[C\cap D] = (c-1)\cdot|C\cap D|$

- ~~Since~~ Let t be the activation of proc. c for which C is critical set
- Since $v[t] > c-1$ we must have $t \notin C\cap D$
- By same argument as in case I we have if $\frac{fv[C\cap D]}{|C\cap D|} \leq c-\delta$

then $\frac{wv[C\cap D]}{|C\cap D|} \geq \delta + 1$

- Now suppose that there is no job j scheduled in C s.t.
  - $S[j, C\cap D] > 0$
- Then $fv[C\setminus D] = v[C\setminus D] > (c-1)\cdot|C\cap D|$ $t\in C\setminus D$
- Hence $\phi(C\setminus D) \geq \phi(C)$ since adding $D\cap C$ to C adds at most $v[C\cap D] = (c-1)\cdot|C\cap D|$ fv
- $C\setminus D$ still fulfills the requirements:
  - $t \in C\setminus D$
  - $fv[C\setminus D] = v[C\setminus D]$
  - $v[t'] \geq c-1 \quad \forall t' \in C\setminus D$

  but has higher density than C ↯ to choice of C

$\Rightarrow \exists j$ scheduled in C s.t. $s[j, C\cap D] > 0$

$\Rightarrow \frac{wv[C\cap D] + s[j, C\cap D]}{|C\cap D|} > \delta - 1$

$\Rightarrow$ there are at least δ jobs with an execution interval intersecting both $C\setminus D$ and $C\cap D$

$\Rightarrow$ these jobs are contained in J_{v_ℓ} or J_{v_r} and V_ℓ or V_r must exist

Realignment of S_{any}:

- $real(S_{any})$:
  - for $k := m$ to 1:
    - $fill(k, [0, d_{max}])$
    - decrement $Res(V)$ for every V s.t. some V' with $V' \cap V \neq \emptyset$ was closed on proc. k

Def Res

- $fill(k, V)$:
  - if $crit(V) \leq 1$: return
  - let C ~~s.t.~~ be critical set s.t. $C \sim V$
  - while $\exists$ (full) active interval $A \subseteq V$ on proc. k s.t.
    - $A \sim V$
    - $\hat{\phi}(A) \leq k-1$ (and hence $\phi(A \cap C) \leq k-1$)
  - do:
    - if ~~Res~~ V_ℓ exists and $Res(V_\ell) \geq 0$:
      - $close(k, V_\ell)$
    - else if V_r exists and $Res(V_r) \geq 0$:
      - $close(k, V_r)$
  - for every valley $V' \subseteq V$ of C which has not been closed on k:
    (V_ℓ, V_r are valleys of C)
    - $fill(k, V')$

- close(k, V):
  - for every $t \in V$ which is idle on proc. k:
    - if proc. $1, \dots, k-1$ idle at t:
      - introduce new dummy active slot on proc. k at t
    - else:
      - move active slot of highest proc. among $1, \dots, k-1$ to proc. k (at t)

$\cdot S_{any}$

- We transform S_{LTR} for the sake of analysis in two steps:

- S_{any}:
  - $\forall t$ of $crit(t) =: k \geq 2$ with $V[t] = k-1$:
    ($V[t] \geq k-1$ by ~~the~~ $crit(t) = k$)

    - Introduce dummy active slot on proc. k

$\Rightarrow$ in S_{any}: $\forall t$ of $crit(t) =: k$:
  - proc. $1, \ldots, k$ active at t

Invariants for real(S_{any}):

At every step in real(S_{any}), it holds for every valley V of crit$(V) \geq 2$

① if $\phi(C \cap D) \leq k_V - \delta$

- for some $\delta \in \mathbb{N}$,
- some interval D spanning C

then
- V_ℓ or V_r exists and
- $Res(V_\ell) + Res(V_r) \geq 2\delta$

② $\forall t \in C \cap V$: proc. $1, \dots, k_V$ active

③ $\forall$ active interval $A \subseteq V$ on k_V with $A \sim V$:

- A spans C

where
- k_V: highest processor s.t.
  - k_V not fully filled yet
  - No $V' \supseteq V$ has been closed on k_V so far
  - $\exists$ full active interval $A \subseteq V$ on k_V

  ($k_V = 0$ if no such proc. exists)

- C: critical set s.t. $C \sim V$

- We call k_V for a specific realignment step the "critical proc." for V

~~I. B.: We must consider~~

- Structured induction over real(S_{avg}):

- I. B.: We must consider S_{avg}:

  - Let V be an arbitrary valley in S_{avg}, C s.t. $C \sim V$, $c = \mathrm{crit}(V)$

    - We first show $k_V \leq C$:
      - otw. V contains full active interval on proc. $k_V > c$
      - hence also an activation $t \in V$ of proc. $k_V > c$

      $\Rightarrow$ by construction of S_{avg}, t must be of $\mathrm{crit}(t) = k_V > c$

      $$↯ \text{ to } \mathrm{crit}(V) = \max_{t \in V} \mathrm{crit}(t) = c$$

Proof of (2): $\forall t \in C$: proc. $1, \dots, k_V, \dots, c$ active

by construction of S_{avg}, ~~choice of C~~, choice of C

Proof of (1):

- Let D be an interval spanning C with
  - $\phi(C \cap D) \leq k_V - \delta \leq c - \delta$

    for some δ

- By Lemma Valley, we have $|\partial V_\ell| + |\partial V_r| \geq \delta$

  ~~$\Rightarrow$ Res~~ and that V_ℓ or V_r exists

  $\Rightarrow Res(V_\ell) + Res(V_r) \geq 2\delta$ initially

Property (3): We show after the induction that (2) $\Rightarrow$ (3)

I.S.: • Suppose properties 1, 2 hold at all steps of $real(S_{avg})$ up to a specific one

• Consider this specific next step ~~and let~~
• Let V be an arbitrary valley of $crit(V) \geq 2$ and k the proc. currently being filled

Case I: Some $V' \supseteq V$ is closed on proc. k

Case II: Some $V' \subset V$ is closed on proc. k

Case III: Some V' s.t. $V' \cap V = \emptyset$ is closed on proc. k

Case IV: $fill(k, [0\ d_{max}])$ returns and $Res(V')$ is decreased by 1 ~~for all V'~~ for $\forall V'$ s.t. some $V'' \cap V' \neq \emptyset$ was closed on k during $fill(k, [0, d_{max}])$

• Let $k_V^{>0}$ be the proc. for V before this next step
k'_V " after "

- Case I: Some $V' \supseteq V$ is closed on proc. k
  $\Rightarrow$ No V'' with $V'' \cap V \neq \emptyset$ has been closed so far on k
  - Also, the stair property holds within V on proc. $1, \dots, k$ before
- We show that $k'_V \leq k_V - 1$

Ia $k_V = k$

- Then $V' \supseteq V$ is closed on k_V
  $\Rightarrow k'_V \leq k_V - 1$ by Def.

Ib $k_V < k$ — since $V' \supseteq V$ is closed on k

- Suppose for c.d. $k_V \leq k'_V < k$
- Let $A \subseteq V$ be a full active interval on k_V before
- We show that $A \subset V$, i.e. there must be some $t \in V$ idle on k_V before the close of V' and hence by stair-prop.:
  - proc. $k_V, \dots, k$ idle at t before
- If some $V' \supseteq V$ is closed, clearly $V \subset [0, d_{max}]$ by Def of real, V_l/V_r / choice of V' as valley of some critical set
  $\Rightarrow \min V - 1 \in [0, d_{max}]$ ← wlog.
  or $\max V + 1 \in [0, d_{max}]$
- We show that $t := \min V - 1$ must be active on k_V before:
  - Let $W \supseteq V$ be valley s.t. $W \sim t$, $t \in W$.
    $\Rightarrow W \supseteq V$ since $W \sim t \succ V$
  - By case assumpt., no $W' \supseteq W (\supseteq V)$ can have been closed on proc. k so far $\Rightarrow k_W \geq k_V$ (before)
    $\overset{\text{I.H.}}{\Rightarrow}$ proc. $1, \dots, k_V, \dots, k_W$ active at t before
    $\Rightarrow$ if $A \overset{\subseteq V}{}$ is full active interval on k_V before, then $\min V \notin A$

• We know by Def. of realignment, ~~closing~~ closing:

$\forall k'$ s.t. $k_V \le k' < k$; $\forall t \in V$:

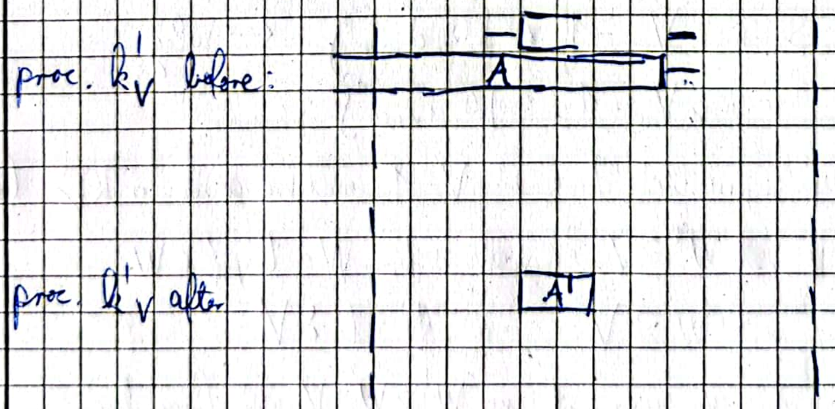
- if t was idle on k' before then t is idle on k' after (Def close, $k' < k$)
- if t was idle on k_V before $\Rightarrow$ t is idle on k' after (stair-prop.)
  - then t was idle on k' before
- if t was part of full active interval $A \subseteq V$ on k_V before,
  then t was idle on $k_V + 1$ before (choice of k_V) ~~stair-prop~~
  then t was idle on k before (stair-prop.)
  then t is idle on k_V after (Def close)

• So for $t \in V$ to be active on k' after,

- t must have been active on k' before (Def close, $k' < k$)
  and t cannot have been part of a full active interval $A \subseteq V$

$\Rightarrow t \in A$ for some <u>partial</u> active interval $A \subseteq V$ on k' before

• For $A' \subseteq V$ to be a full active interval on k'_V after (with $k_V \le k'_V < k$), we must have $A' \subseteq A$



~~$\Rightarrow$ there must have~~

$\Rightarrow$ there must have been an active interval $A'' \subseteq [\min A', \max A] \subseteq V$

on $k'_V + 1 > k_V$ before close

$\lightning$ to ~~the~~ Def of k_V

$\Rightarrow k'_V \leq k_V - 1$

Proof of 1: if $\phi(C \cap D) \leq k'_V - \delta \leq k_V - (\delta + 1)$

then by I. H. V_ℓ or V_r exist and $Res(V_\ell) + Res(V_r) \geq 2(\delta+1)$

(both before and after)

Proof of 2: by I. H.: $\forall t \in C \cap V$: proc. $1, \dots, k_V$ active before

Since at most the uppermost active slot is moved:

$\forall t \in C \cap V$: proc $1, \dots, k_V - 1 \geq k'_V$ active after

- Case II: Some $V' \subset V$ is closed on k
  - Again, no $V'' \supseteq V$ can have been closed on k so far
  - We show that $k_V = k \geq k'_V$
  - Let W be the valley for which V' is closed for, i.e.
    - V' is closed during $\mathrm{fill}(k, W)$

    $\Rightarrow W \supset V'$ $\Rightarrow$ no $W' \supseteq W$ has been closed on k so far
    - Also, there must be some active interval $I \subseteq W$ on k before (Def real)

      $\Rightarrow k_W = k$
    - Since $V' \subset V$ and $V' \subset W$: $V \cap W \neq \emptyset$
      - If $V \prec W$, then by choice of V' as valley of C_W (i.e. max. interval s.t. $V' \prec C_W$), we must have $V \subseteq V'$ ↯

      $\Rightarrow V \nprec W$ and $V \geq W$

      $\Rightarrow k_V \geq k_W = k$

      $\Rightarrow k_V = k \geq k'_V$

Proof of ①: If $\phi(C \cap D) \leq k'_V - \delta \leq k_V - \delta$

$\overset{I.H.}{\Rightarrow}$ V_ℓ or V_r exist and $\mathrm{Res}(V_\ell) + \mathrm{Res}(V_r) \geq 2(\delta + 1)$ (both before and after)

Proof of ②: $V' \cap C = \emptyset$ since $V' \prec C$

$\Rightarrow$ ② follows from I.H., $k'_V \leq k_V$

Case III: some V' s.t. $V' \cap V \neq \emptyset$ is closed on proc. k

- We first show that $\underline{\min V - 1}$, $\max V + 1 \notin V'$

- We consider $\min V - 1 =: t$ ($\max V + 1$ symmetrical)
  - $t \in [0, d_{\max}]$ since V' is valley of some C
  - By choice of V we must have $t \succ V$

  $\Rightarrow$ if $t \in V'$, we would have $V' \succ V$ and hence $V' \supseteq V$ ↯

$\Rightarrow$ schedule is not modified in $[\min V - 1, \max V + 1]$

$\Rightarrow$ no partial active interval in V can become full active interval

$\Rightarrow k_V = k'_V$

① if $\phi(C \cap D) \leq k'_V - \delta = k - \delta$

then by I.H.:
- V_ℓ or V_r exists
- $Res(V_\ell) + Res(V_r) \geq 2\delta$ (before and after)

② Follows directly from I.H.:

$\forall t \in C \cap V$: proc. $1, \dots, k_V = k'_V$ active before

$\Rightarrow$ " " after

Case IV: fill$(k, [0, d_{max}])$ returns and Res is decremented

- Schedule ~~then~~ does not change but proc. k is now fully filled

$\Rightarrow k'_V \leq k_V$

$\Rightarrow$ ② follows from I. H.

IVa · during fill$(k, [0, d_{max}])$, no ~~[illegible]~~ V' s.t. $V' \cap V \neq \phi$ was closed on k

$\Rightarrow$ Res(V) does not change

$\Rightarrow$ If $\phi(C \cap D) \leq k'_V - \delta \quad \leq k_V - \delta$

then V_ℓ or V_r exists

and $Res(V_\ell) + Res(V_r) \geq 2\delta$ (before and after)

IVb · " same "

$\Rightarrow Res(V)$ decremented by 1 to $Res'(V) := Res(V) - 1$

- As seen in cases I, II, III, k_V decreases monotonically during fill$(k, [0, d_{max}])$

- Consider the schedule right before the first $V' \cap V \neq \phi$ was closed on proc. k

- ~~Let k be the cr~~

- Let k^0_V be the critical processor for V at this point before close of V'

- Let k^1_V be the critical proc. " after close of V'

- We have $k'_V \leq k^0_V - 1$:
  - if $V' = V$, then as argued in case I:
    - $k^1_V \leq k^0_V - 1$

    $\Rightarrow k'_V \leq k_V \leq k^1_V \leq k^0_V - 1$
  - if $V' \subset V$, then as argued in case II:
    - $k^0_V = k$

    $\Rightarrow$ since $k'_V \leq k - 1$, (fill(k, [0, dune]) returns)

    we have $k'_V \leq k^0_V - 1$

- By strong I. H.:
  - if $\phi(C \cap D) \leq k'_V - \delta \leq k^0_V - (\delta + 1)$
  - then V_ℓ or V_r exists

  and $Res(V_\ell) + Res(V_r) \geq 2(\delta + 1)$ before

  $\Rightarrow Res'(V_\ell) + Res'(V_r) \geq 2\delta$ after

- We conclude by showing property ③ by prop. ②

③: in every valley V: every active interval $A \subseteq V$ on proc. k_V with $A \sim V$ spans C

- Proof: $A \sim V$ implies that $A \cap C \neq \emptyset$
  - Assume for c.d. A does not span C
    - $\Rightarrow$ $\min A$ is "within" subinterval of C (1)
    - or $\max A$ "
  - wlog (1), i.e. $\min A - 1 \in C$, $\min A \in C$

  Case I: $\min A - 1 \in V$

  ② $\Rightarrow$ $\min A - 1$ is active on k_V

  $\Rightarrow$ A is not a (full) active interval on k_V ↯

  Case II: $\min A - 1 \cancel{\notin W} = \min V - 1$
  - Consider valley W s.t.
    - $\min A - 1 \in W$
    - $\min A - 1 \sim W$

  $\Rightarrow$ $W \supset V$ and $W \succeq V$ and $\min A - 1 \in C_W$

  $\Rightarrow$ $k_W \geq k_V$

  ② $\Rightarrow$ $\min A - 1$ is active on proc. k_V ↯

S_{real}

- $S_{real} := real(\underset{\in V}{S_{avg}})$ is well defined:
  - Since A on k_V always spans C for $A \sim V$, ③
    V_ℓ and V_r are defined ~~$\forall$~~ for A, V in every iteration
  - ~~$\forall$ Since $\forall$ By ①, there is always~~
  - In every iteration, some valley can be closed by ①
  - This reduces the number of ~~active slots~~ idle ~~$\forall$~~ intervals ~~slots~~ on k by ~~$\geq$~~ 1

  $\Rightarrow$ $real(S_{avg})$ ~~$\forall$~~ terminates

~~$\forall$~~

- $\phi(A)$ prop in S_{real}:
  - $\forall k \in [m]$, active interval A on proc. k in S_{real}:

    $$\hat{\phi}(A) > k-1$$

- Proof: • We show ~~show~~ that $fill(k, [0, d_{max}])$ establishes the prop. on k.
  The claim then follows since $fill(k, [0, d_{max}])$ does not change the schedules of proc. above k
  - ~~$\forall$~~ Since in $fill(k, [0, d_{max}])$ we always close valleys for active intervals A on k spanning a corresponding C, we know that ~~on~~ on proc. k, active intervals are only extended.

~~Consider know the point when the while loop during~~

~~fill~~

- Let $A \subseteq V$ be an active interval on proc. k in S_{real} with $A \sim V$, $\mathrm{crit}(A) \geq 2$.

~~Consider the point in fill(k, V)~~

- No $W \supseteq V$ can have been closed on k since otw. there would be no $A \subseteq V$ in S_{real}
- Consider then the point in fill(k, V) when the while-loop terminates
- Clearly at this point all $A' \subseteq V$ of $A' \sim V$ on k have $\hat{\phi}(A') > k-1$
- There must also be at least one such A' at this point for $A \subseteq V$ to be active interval on k in S_{real} with $A \sim V$
- In particular, one such A' must have $A' \subseteq A$

$$\Rightarrow \hat{\phi}(A) > k-1$$

- Lemma "$\forall_i\ crit(A) \leq 1$":
  - For constellations on proc. k in S_{real} of the following form:
    - $[\ A_\ell\ ]$ —— I —— $[\ A_r\ ]$
    - with $crit(A_\ell), crit(A_r) \leq 1$ (and $A_r \neq \emptyset$)
  - We have $\Phi(A_\ell \cup I \cup A_r) \geq 0$
    and $k = 1$
- Proof:
  - Claim: real(S_{avg}) does not create new activation times
    but may only change the proc. activated
    - Formally: t activation of some proc. k in S_{real}
      $\Rightarrow t$ " k' in S_{avg}

$\Rightarrow$ min A_r must have been activation of proc 1 ~~...~~ in S_{avg} since
$crit(A_r) \leq 1$

~~$\Rightarrow$ we show that $k = 1$~~:

$\Rightarrow$ max I is idle in S_{avg} (on all proc.)

$\Rightarrow J_V = \emptyset \ \forall V \ni \max I$ and hence also $Res(V) = 0$

$\Rightarrow$ no V was closed on any proc. s.t. $V \cap [\max I, \max A_r]$

Since A_r is active interval with $crit(A_r) \leq 1$ we must then
have $k = 1$

- For I to be idle on proc. $k = 1$ in S_{real} and $crit(I) \geq 2$,
  some $V \supseteq I$ with $V \cap I \neq \emptyset$ and hence $V \ni \max I$
  would have to be closed ↯

$\Rightarrow crit(I) \leq 1$

$\Rightarrow$ no V with $V \cap [\min A_\ell - 1, \max A_r + 1]$
can have been closed

$\Rightarrow$ the constellation occurs in the same way in S_{avg} on proc. 1:

$\hat{S}_{avg}$: $\boxed{A_\ell}$ — I — j $\boxed{A_r}$ —

- We show that $E_j \subseteq I \cup A_r$ and hence
  $\hat{\phi}(I \cup A_r) \geq 0$

- Otw j could be scheduled at ~~X~~ $\min I$ or $\max A_r + 1$

(Case I): $\min I \in E_j$ $(\min I - 1 \in E_j)$

$\Rightarrow$ The Algorithm would have clearly extended A_ℓ by scheduling j at $\min I$ instead of $\min A_r$ ↯

(Case II): $\max A_r + 1 \in E_j$

$\Rightarrow$ The Algorithm would have clearly ~~ex~~ extended ~~X~~ I by scheduling j at $\max A_r + 1$ instead of $\min A_r$ ↯

Proof of initial claim:

- Consider the first step in ~~[illegible]~~ real(S_{avg}) in which some t becomes an activation of some proc. k' where t was no activation of any proc. before
- This step must be the closing of some valley V on some proc. $k > k'$:
  - on k, we have seen that the closing of some valley can only merge active intervals
  - ~~[illegible]~~ on proc. $> k$, the schedule does not change

- On $k'' < k$, active slots are only removed
  $\Rightarrow$ $t - 1$ must have been active before on k' and idle on $k'+1, \dots, k$

$s^{k'}$ after:
s^{k} before
$s^{k'}$ before

Case I: $t \in V$
$\Rightarrow k'+1$ (or k) must have been active before at t
$\Rightarrow t$ was activation before ↯

Case II: $t \notin V$
$\Rightarrow t \succ V$

- Let W be the valley s.t. V is closed in fill(k, W)
  $\Rightarrow W \supset V$
- If $t \in W$, then $t \sim C_W$, $t \in C_W$
  $\Rightarrow$ proc. $1, \dots, k_W = k$ active at t before
  $\Rightarrow t$ activation before ↯
- If $t \notin W$, then let W' be the valley s.t. $t \sim W'$, $t \in W'$
  $\Rightarrow W < t \sim W'$ and $W' \supset W$ and $t \in C_{W'}$
  $\Rightarrow k_{W'} \geq k_W = k$
  $\Rightarrow$ proc. $1, \dots, k$ active at t before
  $\Rightarrow t$ activation before ↯

- In total, we have established for every idle interval I on proc. k in S_{real}

$$[\ A_l\ ] \overset{I}{\text{———}} [\ A_r\ ] \quad \text{with } A_r \neq \phi \text{, that}$$

$$\hat{\phi}(A_l \cup I \cup A_r) \geq k-1$$

$\Rightarrow$ For every $k \in [m]$, every I in S^k_{opt} ending ~~at~~ before d_{max}:

I intersects at most 2 distinct idle intervals in S_{real} (*)

- In particular, this holds for every idle interval which is off in S_{opt}

- ~~k~~ $\text{costs}(S_{real}) \leq 2\,OPT + P$:

- Claim: $\text{idle}(S^k_{real}) \leq 2\,\text{off}(S^k_{opt}) + \text{on}(S^k_{opt}) \quad \forall k \in [m]$
  - Let $\mathcal{I}_1$ be set of idle intervals on S^k real intersecting some off-interval of S^k_{opt}
  - By (*), we have $|\mathcal{I}| \leq \#$ off-intervals in S_{opt} $\cdot 2$
    $\Rightarrow \sum_{I \in \mathcal{I}_1} \min\{|I|, Q\} \leq 2\,\text{off}(S^k_{opt})$
  - Let $\mathcal{I}_2$ be set of idle intervals on S^k real not intersecting any off interval in S^k_{opt}
    $\Rightarrow \sum_{I \in \mathcal{I}_2} |I| \leq \text{on}(S^k_{opt})$

(*) What about $I \in \mathcal{I}_2$ overlapping off-interval in S^k_{opt} ending at d_{max}? ~~See S. Notes about~~ Not possible, since I cannot end at d_{max}

\# What about $\text{cnt}(A) \leq 1$?

- Claim: $\text{active}(S_{real}) \leq 2P$
  - By construction of S_{aug} and the Definition of Res, close, we introduce at most as many ~~idle~~ dummy active slots at every t, as there are jobs scheduled at t in S_{PLTR}:
    - for S_{aug}, auxiliary dummy slot is only added for t w/ $\text{cnt}(t) \geq 2$ and hence $V[t] \geq 1$
    - $\text{Res}(V) = 2|J_V|$ initially and is decremented whenever some $t \in V$ is closed
    - Auxiliary active slots ~~are also used in~~ from S_{aug} are used in real

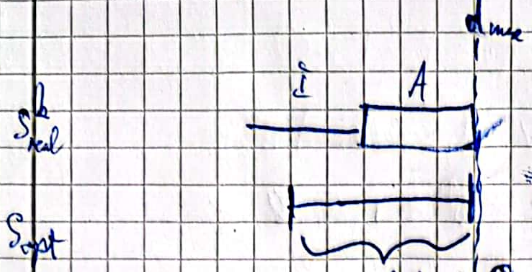
$$\Rightarrow cost(S_{real}) \le 2 off(S_{opt}) + on(S_{opt}) + 2P$$
$$\le 2\,OPT + P$$

- $cost(S_{LTR}) \le cost(S_{real})$:
  - ~~Remo~~
  - Transform S_{real} back into S_{LTR} without decreasing costs:
    - Removing dummy active slots: ✓
      - does not decrease costs
    - ~~A~~ $\forall t$: move active slots down to lower processors:
      - does not increase costs (same argument as for $\exists$ of S_{opt} using lower proc. first
    - real(S_{any}) only moved scheduled jobs to different processors, but not different slots

  $\Rightarrow$ resulting schedule is S_{LTR} up to permutations of jobs at fixed t's across active processors

  $\Rightarrow$ same costs as S_{LTR}

$\bigcirc$ 03.02.23



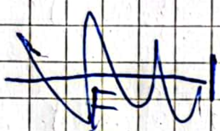
cost of O !

I in S^1_{real}
intersecting $[t, d_{max}]$ in S^1_{opt}

"Kätten on nor off" (i.e. $|I|$ not bounded by on(S^k_{opt}))

- Only possible for $k = 1$.
  otw. $\phi(A) \geq k - 1$

Z.z.: (1) idle interval $[t, d_{max}]$ in S^1_{opt} intersects at most 1 idle interval in S^1_{real}



(2) idle interval $[0, t]$ in S^1_{opt} intersects at most 1 idle interval in S^1_{real}

=> # idle intervals in S^1_{real} intersecting some off-interval in S^1_{opt} is

$\leq 2\,|\mathcal{F}^1_{opt}| - 2$

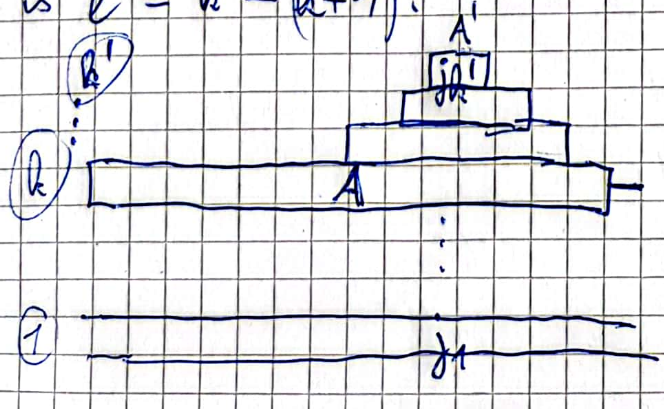
≤ 2 off

Running time of PLTR

02.02.23

- Running time: $(n+m) \cdot \log(d_{max}) \cdot F$
  where F: time for flow calculation

- Proof: - We bound the number of active intervals across all proc. in S_{PLTR} by n
  - This then bounds the number of calls to keepidle and keepactive by $n+m$
  - We derive the bound of n by constructing an injective mapping f from the set of active intervals to J
  - We consider the active intervals in the same order as PLTR, i.e. from „Top-Left to Bottom-Right" and construct f s.t. $f(A) = j \Rightarrow d_j \in A$
- Suppose we have constructed such a (partial) mapping for active intervals on proc. $m, \dots, k$ up to some active interval A on k
- Let A' ~~[illegible]~~ be the last plateau s.t. $A' \subseteq A$,
  let $k' \geq k$ be the proc. of this plateau
- By construction of f and the choice of A', there are at most $k' - (k+1)$ distinct jobs j with $d_j \in [\min A', \max A]$ already mapped to by f
  - This is since the number l of active intervals A'' on proc. $k+1, \dots, m$ s.t. $A'' \cap [\min A', \cancel{\min} \max A]$ is $l \leq k' - (k+1)$:



- Let C_t be the critical set over activation $t := \min A'$ of proc. k'
- Let $J' = \{j_1, \dots, j_{k'}\}$ be the k' distinct jobs scheduled at t

$\Rightarrow$ $\max A + 1 \notin C_t$ since $v[\max A + 1] < k \le k'$ and $\max A + 1 > t$

$\Rightarrow$ $\forall j \in J'$ with $d_j \geq \max A + 1$ is scheduled at $\max A + 1$

$\Rightarrow$ There are at least $k' - (k-1)$ distinct $j \in J'$ with $d_j \in [\min A', \max A]$

$\Rightarrow$ one of those is not mapped to by f so far

$\Rightarrow$ set $f(A)$ to this job

Running time Antoniadis 6-approx:

- Calculating $\ell([t, t'])$'s:

  $O(F\ n^2 \log n)$

- Computing min-cost skeletons $\forall k \in [n]$?

- Combining them for parallel skeleton ?

- Extending parallel min-cost skeleton to feasible sol:

  P · F
  up

  each extension reduces max-def by one

=> only pseudo-polyn.

  & Anton. does not claim smth. 6+ε approx.