PLTR

Algorithm:

~~pltr(q, t, S):~~

- ~~$k = m$~~
- ~~$t = 0$~~
- ~~while $k > 0$:~~
  - ~~while $t \leq d_{max}$~~

- initially $l_t = 0$, $m_t = m$ $\forall t$
- pltr(q, t):
  - if $q = 0$: return
  - ~~if $t = D$ return pltr$(q-1, 0)$~~
  - ~~t = keepidle(q, t)~~
  - ~~t'' = keepactive(q, t)~~
  - while $t < D$:
    - t = keepidle(q, t)
    - t = keepactive(q, t)
  - return pltr$(q-1, 0)$

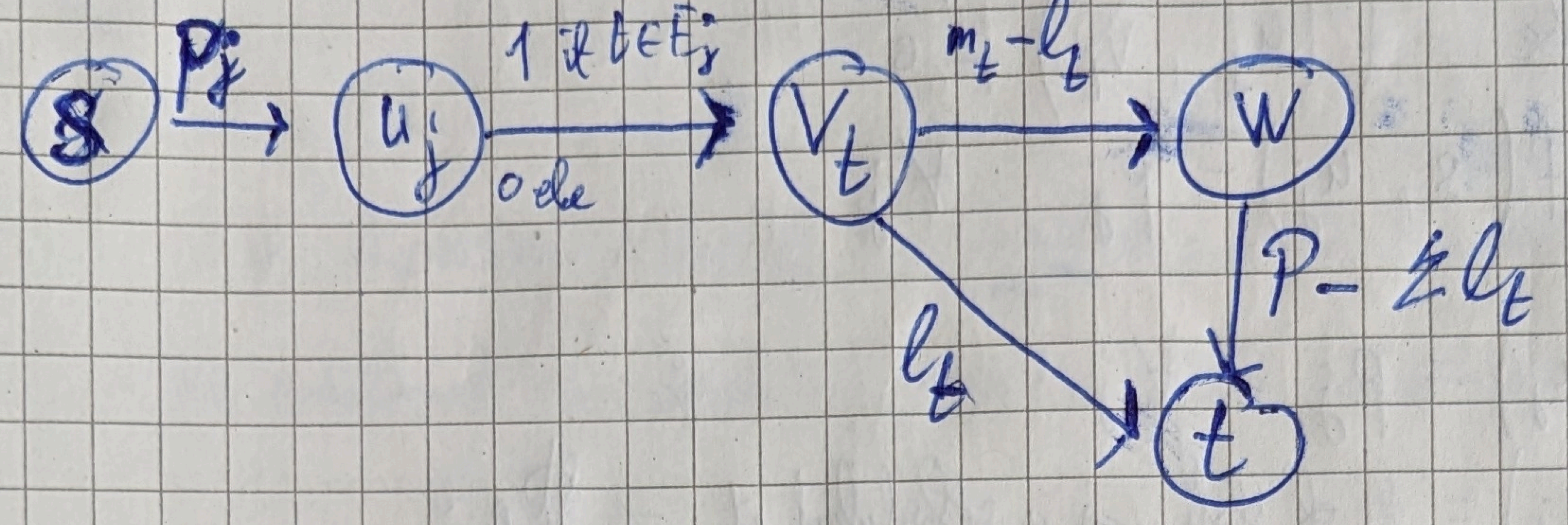
keepidle(q, t):

- search for maximal $t' \geq t$ s.t. there is a feasible schedule with $m'_{t''} = q-1 \quad \forall t'' \in [t, t')$
- set $m_{t''} := q-1$ for all these t''
- return t'

keepactive(q, t):

- search for maximal $t' \geq t$ s.t. there is a feasible schedule with $l'_{t''} = \max\{q, l_{t''}\} \quad \forall t'' \in [t, t')$
- set $l_{t''} =$ [illegible] for all these t''
- return t'

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



- Exists feasible ~~[illegible]~~ schedule respecting l_t, m_t $\Leftrightarrow$ max flow $= P$

~~A~~ $\Leftarrow$:

- Let f be a s-t flow of $v(f) = P \in$ ~~[illegible]~~ $\mathbb{N}_0$
- 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 ~~[illegible]~~, m_t:
    - $c(v_t, w) + c(v_t, t) \leq m_t - l_t + l_t$

      flow conserv. $\Rightarrow$ ~~[illegible]~~ $\forall t$: at most m_t jobs scheduled at t

  - Respects l_t:

    ~~$c(v_t, \ldots) \ldots (\{s\} \cup \{u_j\} \cup \{v_t\} \setminus \{w\})$~~

    - $c(V\setminus\{t\}, \{t\}) = P$

      $\Rightarrow f(v_t, t) = l_t \quad \forall t$

      $\Rightarrow$ ~~[illegible]~~ By flow conservation: $f_{in}(v_t) \geq l_t \quad \forall t$

      $\Rightarrow \forall t$: $\geq l_t$ jobs scheduled at t

$\Rightarrow$ 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 $= p_j$ (fea)

- 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 $f(w, t) = P - \sum l_t$

  - $f_{in}(w) = \sum_t f_{in}(v_t) - l_t$

    $= \sum_t \#$ jobs scheduled at $t - \sum_t l_t$

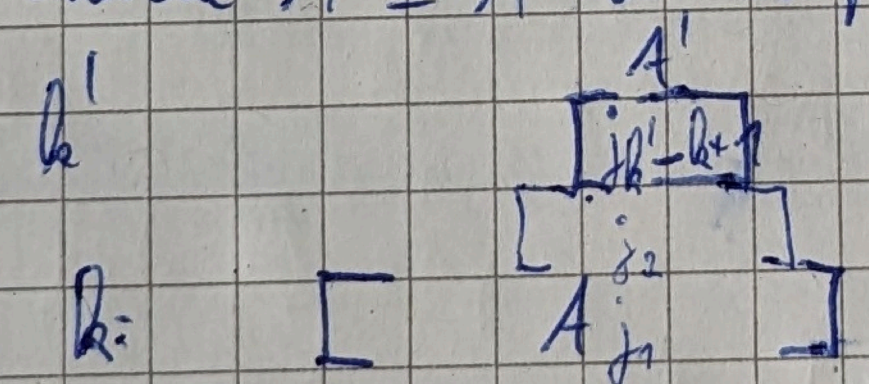
    $\overset{\text{schedule feasible}}{=} P - \sum_t l_t = f_{out}(w)$

- Correctness / feasibility of PLTR:
  - if instance feasible, then l_t, m_t is only changed s.t. there still exists a feasible schedule

- Running time:

  # of keepactives / keepidles (over all proc.) is bounded by n

  - We construct an injective mapping f from active intervals to jobs (s.t. $d_j \in A$)
  - Start at the first, i.e. earliest, active interval A on the highest proc. (wlog m)
    - there is some job j s.t. $d_j \in A$:
      - otw. schedule some job scheduled at $\min A$ at $\max A + 1$ instead on m.
        $\Rightarrow$ still feasible but delayed activation $\lightning$
    - $f(A) := j$
- Suppose we have constructed such a mapping for active intervals on proc. $m, \dots, k$ up to active interval A
- (Let k' with $k \leq k' \leq m$ be highest proc. s.t. there is an active interval $A' \subseteq A$ on this proc. and let A' be the last such interval on k'



)

- ~~Let $A' \subseteq A$ be the active interval on proc. $k' \geq k$~~
- Let $A' \subseteq A$ be the last plateau, $k' \geq k$ the proc. which this plateau is on
- By construction, there are at most $k' - (k + 1)$ jobs j with $d_j \in A'$ already mapped to (and stair-property)
- Suppose there are not at least $k' - k$ such jobs
  $\Rightarrow \exists j \in \{j_1, \dots, j_{k'-k+1}\}$ s.t. $d_j > \max A'$
  $\Rightarrow$ Alg would have delayed A' on k' by scheduling j at $\max A' + 1$ on some proc. $\leq k'$ instead of at $\min A'$ $\lightning$

$\Rightarrow \exists$ some job j with $d_j \in A' \subseteq A$ not yet mapped so far
$\Rightarrow$ define $f(A) = j$

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
  - $w_S[j, Q] = v_S[j, Q] - fv[j, Q]$
  - $pv[j, Q] = \min\{|E_j \cap Q|, p_j\}$

    $\# \; pv[j, Q] \geq v[j, Q] \geq fv[j, Q]$
  - $V[J', Q] := \sum_{j \in J'} v[j, Q]$
  - $s_S[j, Q] =$ # of slots in $Q \cap E_j$ which j is not scheduled in for S
  - $\phi(Q) = \frac{fv[J, Q]}{|Q|}$
  - $\hat{\phi}(Q) = \max_{Q' \subseteq Q} \phi(Q')$
  - $v[Q] := v[J, Q]$

(with ℓ_t, m_t)

Lemma: feasibility characterisation with def, excess

- Definition:
  - $\mathrm{def}(Q) := \mathrm{pr}[Q] - \sum_{t \in Q} m_t$
  - $\mathrm{exc}(Q) := \sum_{t \in Q} \ell_t - \mathrm{pr}[Q]$
- Lemma: an instance with ℓ_t, m_t is feasible

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

- "$\Rightarrow$" trivial

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

  if $\mathrm{def}(Q) > 0$ " m_t "

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

~~We show that def~~ /

$\Rightarrow$ corresponding max flow f has value $v(f) < P$

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

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

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

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

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

~~Case I~~ $\Rightarrow$

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. $t \in E_j$ $= \lvert E_j \setminus Q(S) \rvert \overset{(*)}{\geq} p_j - \mathrm{fo}[j, Q(S)]$
v_t	$l_t + m_t - l_t = m_t$

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

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

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

$$\geq P - \mathrm{def}(Q(S))$$

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 $\text{excess}(Q) > 0$ for some Q

Lemma critical sets (of timeslots)

- For every activation slot t of proc. k in S_{PLTR}:
  - $\exists Q$ s.t.
    - $t \in Q$
    - $fr[Q] = v[Q]$
    - $\forall t' \in Q$: proc. $1, \dots, k-1$ active at t'

Proof: Suppose for c.d. that there is some activation t of proc. k s.t. no such Q exists

$\Rightarrow \forall Q$ s.t.
- $t \in Q$
- $\forall t' \in Q$: proc. $1, \dots, k-1$ active $\circledast$

we have $v[Q] > fr[Q]$

- We show that PLTR would have extended the idle interval on proc. k which ends at t
- ~~Ind~~ For the l_t's, m_t's returned by ~~AS~~ pltr, make the following modifications:
  ~~decrease~~
  - decrease m_t by one
  - increase $m_{t'}$ by one $\forall t'$ with $m_{t'} < k-1$
  - Every Q s.t. $\exists t' \in Q$ with proc. $k-1$ idle and $t \in Q$ has $m_{t'} < k-1$
    $\Rightarrow$ def(Q) does not decrease
  - Every Q s.t. $\circledast$ has def$(Q) < 0$ before the change
    $\Rightarrow$ def$(Q) \leq 0$ after change

$\Rightarrow$ instance still feasible

At time when keepidle returns t for proc. k in pltr.:

m_t's only more permissive (i.e. they monotonically decrease) $\Rightarrow$ delay t

- We call such Q over activation t on proc. k
  tight set ~~over~~ Q_t over ~~x~~ "

- Critical set C_t over "
  is maximum of $\{Q_t\}$ in regard to ϕ

  i.e. $$C_t := \arg\max_{\substack{Q \text{ s.t.} \\ t \in Q; \\ fv[Q] = v[Q); \\ \forall t' \in Q:\, v[t'] \geq k-1}} \{\phi(Q\}$$

- As this set of tight Q_t is clearly closed under union,
  for the sake of uniqueness we take C_t as inclusion maximal

- Definitions based on C_t
  - total order $\succsim$ 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 $\succsim$, i.e.
    $$\mathrm{rank}(C_t) \geq \mathrm{rank}(C_{t'}) \Leftrightarrow C_t \succsim C_{t'}$$

~~# drop subscripts, just C now~~

  - $\mathrm{crit}(C_t) = c \Leftrightarrow t$ is activation of proc. c
    (c is the highest proc. activated at t)

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

    - Interval D:
      - $\mathrm{rank}(D) = \max\{\mathrm{rank}(t) \mid t \in D\}$
      - $\mathrm{crit}(D) =$ " crit "

  - An interval V is called a valley on $\mathrm{rank}(C)$ if
    - $C \sim V$
    - V is maximal (in regard to "$\supseteq$")

  - Jobs:
    $$\mathrm{rank}(j) = \max_{t:\, j \text{ is scheduled at } t} \mathrm{rank}(t)$$
    $$\mathrm{crit}(j) = \text{ " } \mathrm{crit}(t)$$

~~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 $\min C \cap D - 1$
    - 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:

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

$$\text{then } |J_{V_\ell}| + |J_{V_r}| \geq \delta$$

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: ~~$v[C] \geq$~~ $v[C \cap D] \geq$ ~~c~~ $(c-1) \cdot |C \cap D|$

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

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

$\Rightarrow$ 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}

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 $v[t] \geq c-1$ 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.

• $S[j, C \cap D] > 0$ $\quad t \in C \setminus D$

• Then $fv[C \setminus D] \geq v[C \setminus D] > (c-1) \cdot |C \cap D|$

• 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 $\lightning$ 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 ~~one~~ 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}