Overview

- Background / motivation (Nico's visit & using SOCO / CFC results towards applications for cloud computing or energy systems)
- Geographic Load Balancing problem
- Carbon-intelligent Pause & Resume problem
- Discussion

SOCO / Convex Function Chasing (bkgd)

- Problem: choose decisions x_t online inside a normed vector space to minimize total hitting + switching cost

$$\min \sum_{t=1}^{T} f_t(x_t) + ||x_t - x_{t-1}||$$

- Hitting cost: convex functions arriving online $f_t(x_t)$
- Switching cost: “distance” between x_t and x_{t-1}.
- Learning-augmented results [Christianson, Handina, Wierman ‘22]
  - $(1+\epsilon)$**-consistency** and $O\left(\frac{CD}{\epsilon}\right)$-robustness — significant improvement on online algorithms without advice for this problem setting.

Carbon-Intelligent Pause and Resume

- We have a job of length k, and we want to selectively run the job to minimize the carbon cost, while finishing the job before deadline T.
- *Formulation:* ($x_t \in \{0,1\}$, β is "pause/resume" cost)

$$\min \sum_{t=1}^{T} c_t x_t + \beta \| x_t - x_{t-1} \|, \quad \text{s.t.} \sum_{t=1}^{T} x_t \geq k$$

  - Adjacent to convex function chasing and k-min search
  - Compared to CFC we have this added deadline constraint
  - Compared to k-min search we have this added switching cost of changing the decision across time steps.

Pause and Resume (as a 1-min problem)

- 1-min Search Review:
  - Optimal threshold-based algorithm [Sun et al. '21, El-Yaniv '01]
    - Accept first item with value below $\sqrt{LU}$, where all item values are bounded by the interval $[L, U]$. $\sqrt{\frac{U}{L}}$ -competitive.

- How to modify this threshold-based algorithm for our carbon-intelligent pause and resume problem?
  - Need to consider switching cost of pausing or resuming
  - (also, what about k-min search instead? more on this later....)

Pause and Resume (as a 1-min problem)

- **Idea:** Use *two* thresholds for hysteresis. Call these h_1 and h_2. Then our *state changes* depend on the current state ($x_t \in \{0,1\}$)
  - if $x_t = 1$, i.e. we are currently processing the job, wait until carbon costs rise above h_2 to stop processing it.
  - if $x_t = 0$, i.e. we are not processing the job, wait until carbon costs fall below h_1 to start processing it.
- What should the actual value of h_1 and h_2 be?

Pause and Resume (as a 1-min problem)

- What should the actual value of h_1 and h_2 be?
  - (skipping some steps) analyze possible input cases w.r.t. competitive ratio in terms of h_1 and h_2.
  - Result:
    - $h_1 = \dfrac{\sqrt{\beta^2 + 4UL} - \beta}{2}, \qquad h_2 = \dfrac{\sqrt{\beta^2 + 4UL} - \beta}{2} + \dfrac{k\beta}{(k-1)}$
    - $\dfrac{\text{ALG}}{\text{OPT}} = \max\left\{\dfrac{U}{h_1}, \dfrac{h_2}{L}\right\} \approx \sqrt{UL} + O\left(\dfrac{\beta}{L}\right)$

Pause and Resume *(as a k-min problem)*

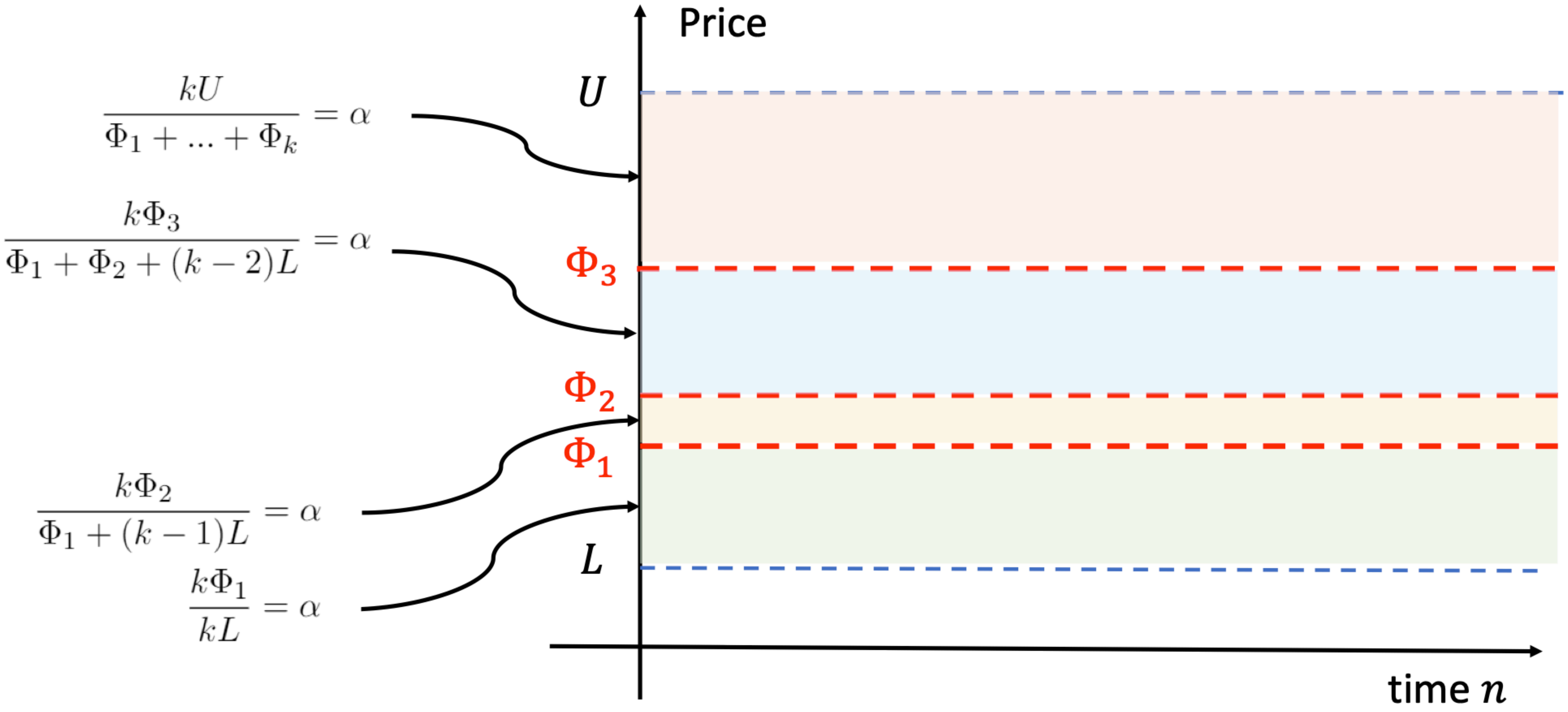
- Initially avoided using k-min search because of the "changing threshold" interfering with switching cost [Sun et al. '21, El-Yaniv '01]
- **Idea:** Use the same idea of hysteresis, but for k-min search moving threshold values.
  - example: if we're processing a job and the threshold is below the current carbon cost, continue processing the job as long as the cost is not more than β + current threshold (or a similar offset).

Pause and Resume *(as a k-min problem)*

- Existing algorithm for k-max search [Sun et al. '21, El-Yaniv '01]
  - Let $\theta = \dfrac{U}{L}$ Obtain α by solving: $\dfrac{\theta - 1}{\alpha - 1} = \left(1 + \dfrac{\alpha}{k}\right)^k$
  - This gives an α-competitive algorithm for k-max, where we have a family of k thresholds:
    - Threshold for the ith unit is $\Phi_i = L\left[1 + (\alpha - 1)\left(1 + \dfrac{\alpha}{k}\right)^{i-1}\right]$

[Sun et al. ’21]

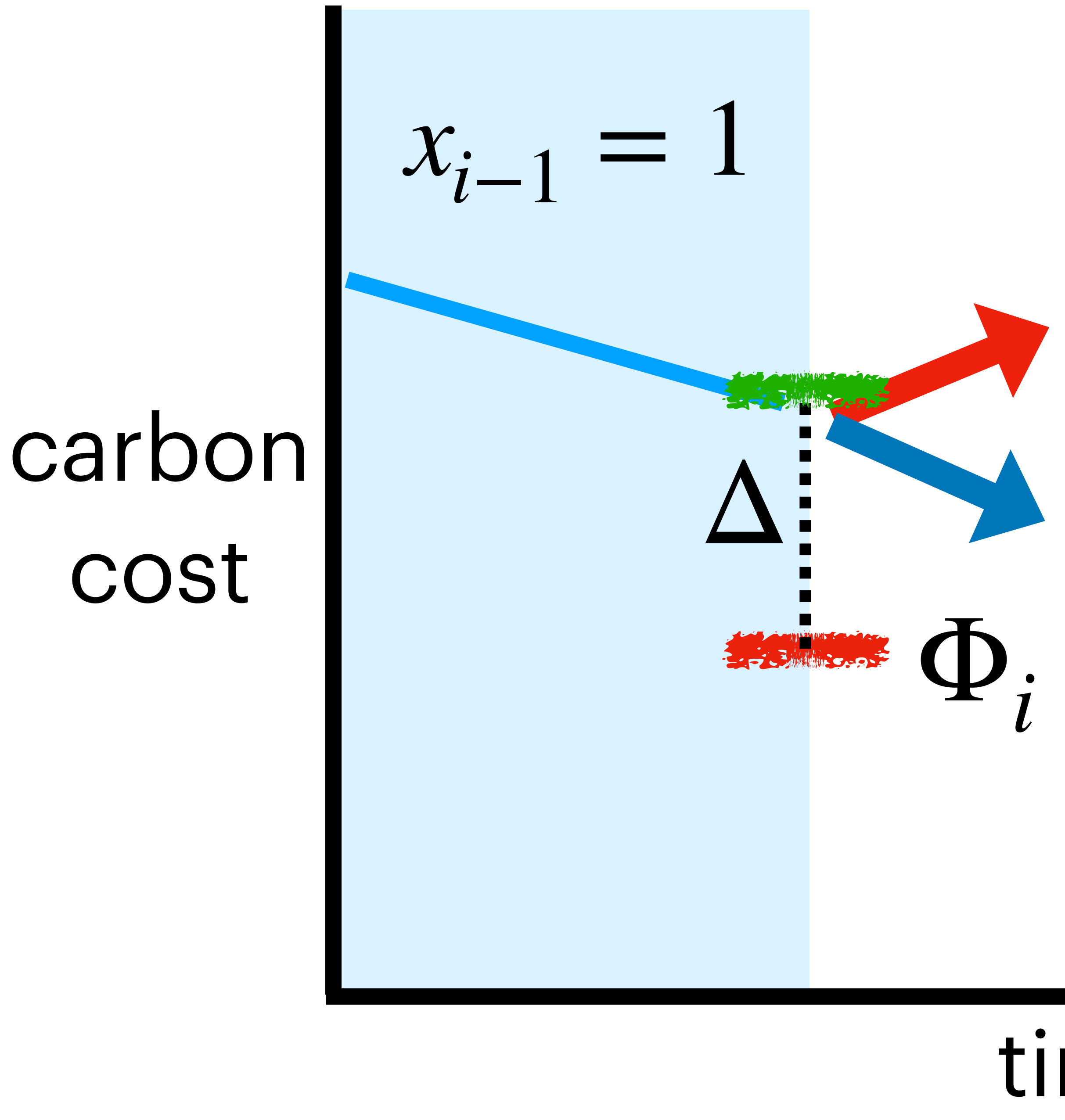
Balancing Rule in k-Max Search



Pause and Resume (*as a k-min problem*)

- How to incorporate *buffer* into these thresholds which captures the switching cost?

- Warmup: consider a single time slot, where we have some threshold and a yet unspecified buffer.



When is it better to keep processing? Consider future potential cost.

$$\frac{ALG}{OPT} = \frac{(k-i)U + 2\beta}{\Phi_i} = \frac{(k-i-1)U + \Phi_i + \Delta}{\Phi_i}$$

$$= \frac{\sum_{t=i}^{k} \Phi_t + 2\beta}{L} = \frac{\sum_{t=i}^{k} \Phi_t + \Delta}{L}$$

$$\Phi_i + \Delta \leq 2\beta + U$$

$$\Delta \leq 2\beta$$

Pause and Resume (*as a k-min problem*)

- How to incorporate *buffer* into these thresholds which captures the switching cost?

More concretely, for any arbitrary time slot *i*:

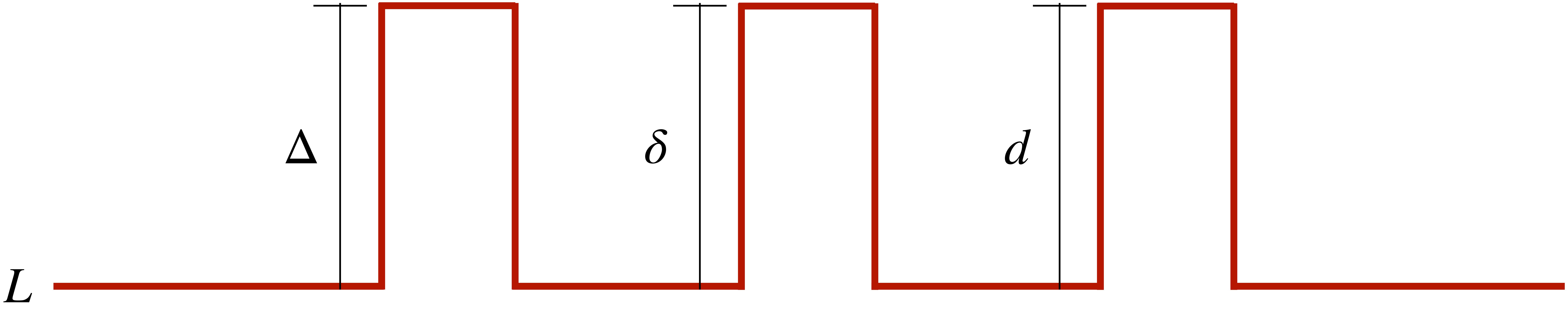
$$\frac{ALG}{OPT} = \frac{\sum_{t=0}^{i-1} \Phi_t + (k-i)U + 2\beta}{iL + (k-i)\Phi_i} = \frac{\sum_{t=0}^{i} \Phi_t + (k-i-1)U + \Delta}{iL + (k-i)\Phi_i}$$

$$= \frac{\sum_{t=1}^{k} \Phi_t + 2\beta}{L} = \frac{\sum_{t=1}^{k} \Phi_t + \Delta}{L}$$

How does this generalize to the case where multiple consecutive time slots are above the threshold?

Intuition: what would *OPT* do?

- Consider a simplified example. Here is some carbon cost arriving online:



$kL + \Delta + \delta + d$

(no switch)

$kL + 2\beta + \delta + d$

$kL + 2\beta + \Delta + d$

$kL + 2\beta + \Delta + \delta$

$kL + 4\beta + \delta$

$kL + 4\beta + \Delta$

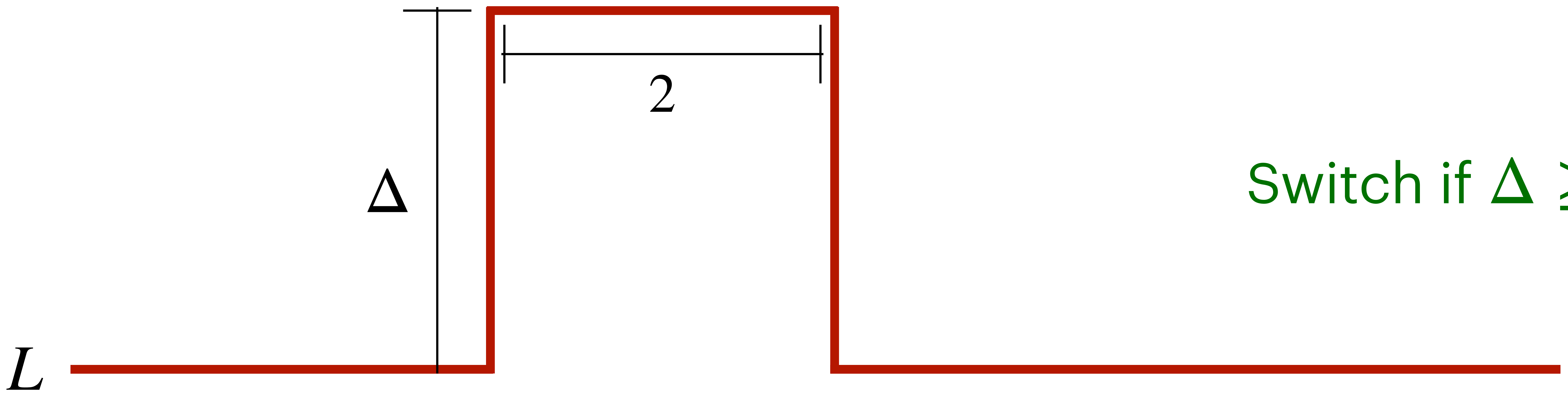
$kL + 4\beta + d$

$kL + 6\beta$

(switch each time)

Intuition: what would *OPT* do?

- Different simplified example. Here is some carbon cost arriving online:



Switch if $\Delta \geq \beta$

$kL + 2\Delta$

$kL + 2\beta + \Delta$

$kL + \Delta + 2\beta$

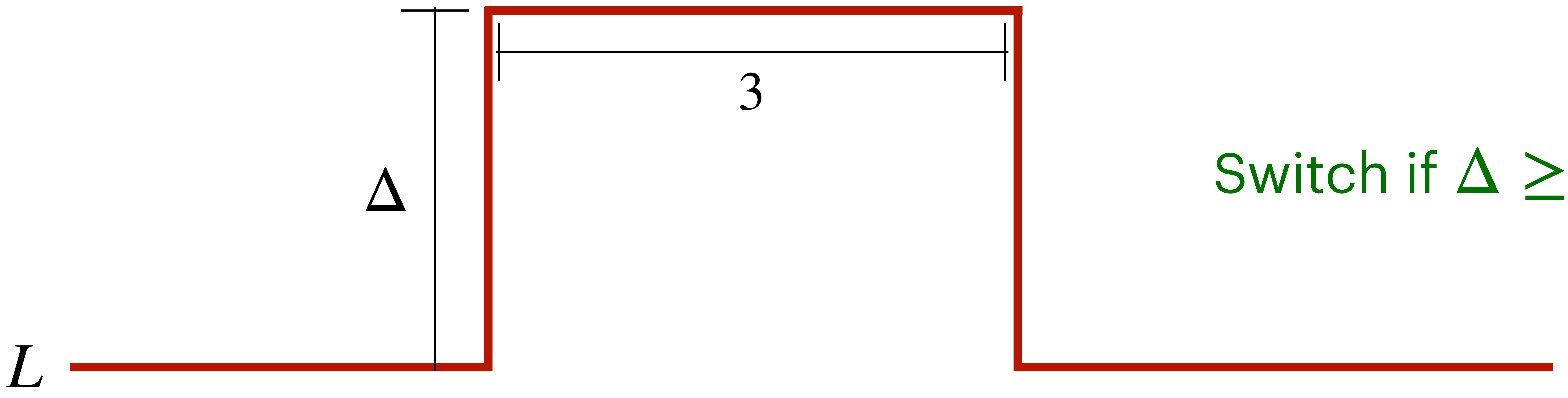
$kL + 2\beta$

(no switch)

(switch once)

Intuition: what would *OPT* do?

- Different simplified example. Here is some carbon cost arriving online:



Switch if $\Delta \geq \frac{2}{3}\beta$

$kL + 3\Delta$

$kL + 2\beta + 2\Delta$

$kL + 2\Delta + 2\beta$

$kL + \Delta + 2\beta$

$kL + 2\beta + \Delta$

$kL + 2\beta$

(no switch)

(switch once)

Back to Pause & Resume *(as a k-min problem)*

- How to incorporate *buffer* into these thresholds which captures the switching cost?

We had the following for a single time slot i:

$$\frac{ALG}{OPT} = \frac{\sum_{t=1}^{i-1} \Phi_t + (k-i)U + 2\beta}{iL + (k-i)\Phi_i} = \frac{\sum_{t=1}^{i} \Phi_t + (k-i-1)U + \Delta}{iL + (k-i)\Phi_i}$$

$$= \frac{\sum_{t=1}^{k} \Phi_t + 2\beta}{L} = \frac{\sum_{t=1}^{k} \Phi_t + \Delta}{L}$$

Idea: When $x_t = 1$ and $x_{t-1} = 0$, set $E = 0$.

If there is some *excess*, i.e. non-zero Δ above Φ_i,

set $E = E + \Delta$ and keep processing as long as $E \leq 2\beta$

k-min with buffer

Competitive Analysis - k-min w/ buffer

- Consider the sequence of items that gives exactly the optimal competitive ratio for k-min:

$$\sigma = \langle \Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_k, \{L\}^k, \{U\}^k \rangle$$

$$\frac{ALG}{OPT} = \frac{\sum_{i=1}^{k} \Phi_i}{kL} = \alpha$$

Competitive Analysis - k-min w/ buffer

- Consider the sequence of items that gives the exact competitive ratio for k-min — ***but this time we introduce bad items for switching***

$$\sigma = \langle \Phi_1, U, \Phi_2, U, \Phi_3, \dots, U, \Phi_k, \{L\}^k, \{U\}^k \rangle$$

standard k-min

$$\frac{ALG}{OPT} = \frac{\sum_{i=1}^{k} \Phi_i + k2\beta}{kL} = \alpha + \frac{2\beta}{L}$$

k-min with buffer

$$\frac{ALG}{OPT} = \frac{\sum_{i=1}^{k} \Phi_i + k2\beta}{kL} = \alpha + \frac{2\beta}{L}$$

(potential problem)

Competitive Analysis - k-min w/ buffer

- Consider this sequence of items — ***some good news?***

$$\sigma = \langle \Phi_1, \Phi_2, \Phi_3, \ldots, (\Phi_{k-2} + \beta/3), (\Phi_{k-1} + \beta/3), (\Phi_k + \beta/3), \{U\}^k \rangle$$

standard k-min

$$\frac{ALG}{OPT} = \frac{\sum_{i=1}^{k-3} \Phi_i + 3U}{\sum_{i=1}^{k} \Phi_i + \beta}$$

k-min with buffer

$$\frac{ALG}{OPT} = \frac{\sum_{i=1}^{k} \Phi_i + \beta}{\sum_{i=1}^{k} \Phi_i + \beta} = 1$$

better! so the worst-case is the same, but in practice we should do better.

Final Comments

- When $L = 0$, problem seems difficult — e.g. competitive ratio often becomes undefined.
- $\dfrac{\beta}{L}$ has made an appearance in several places, can we derive a lower bound on online search problem + switching cost of β?
- What if β is large? Trivial lower bound when $L + \beta \geq U$
- ***next step:***
    - "rederive" the thresholds $\{\Phi_i\}_{i \in [1,k]}$ [Sun et al. '21, El-Yaniv '01], *this time incorporating the buffer in analysis*

Looking Ahead

- Given results, thinking about submitting this somewhere
  - *1-min results & analysis*
  - *k-min results & analysis*
  - numerical experiments
    - on real carbon traces, vs. existing / “naive” algorithms
  - extend to one-way trading?
  - learning-augmented versions

e-Energy - Feb 3 **SIGMETRICS - Feb 1** **ICML - Jan 26**

"Lower Bound" for k-min w/ switching cost

- Consider the sequence:

$$\sigma = \langle U, L, U, L, U, L, \ldots, \begin{cases} \rightarrow U, U, U, U, U \ldots & \textbf{case 1} \\ \rightarrow L, L, L, L, L \ldots & \textbf{case 2} \end{cases}$$

- Suppose that ALG switches s times.

$$\frac{ALG}{OPT} \geq \min_s \max \left\{ \frac{2(s+1)\beta + sL + (k-s)U}{k2\beta + kL}, \frac{2(s+1)\beta + kL}{2\beta + kL} \right\}$$

- solve for s:

$$\frac{2(s+1)\beta + sL + (k-s)U}{k2\beta + kL} = \frac{2(s+1)\beta + kL}{2\beta + kL}$$

Lower Bound Idea (cont.)

$$\frac{2(s+1)\beta + sL + (k-s)U}{k2\beta + kL} = \frac{2(s+1)\beta + kL}{2\beta + kL}$$

$$\frac{2(s+1)\beta + (k-s)U}{k2\beta} = \frac{2(s+1)\beta}{2\beta}$$

$$2(s+1)\beta + (k-s)U = (s+1)k2\beta$$

$$2s(k-1)\beta + sU = kU + 2(k-1)\beta$$

$$s = \frac{kU + 2(k-1)\beta}{2(k-1)\beta + U} \approx O(k) \qquad \beta = \frac{U}{2(k-1)} \qquad \rightarrow \frac{kU+U}{2U} = \frac{k}{2}$$

when β small

Lower Bound Idea (cont.)

(implicit assumption on OPT)

$$kL + k2\beta \leq kU + 2\beta$$

$$L + 2\beta \leq U + \frac{2\beta}{k}$$

$$\beta = O\left(\frac{U}{k}\right)$$

$$\frac{2(s+1)\beta + sL + (k-s)U}{k2\beta + kL}$$

$$\frac{2(s+1)\beta + kL}{2\beta + kL}$$

$$\frac{U + sL + (k-s)U}{2U + kL} \approx O(k)$$

$$\frac{U + kL}{2\frac{U}{k} + kL} \approx O(k)$$

Re-deriving k-min thresholds

- Existing *k* thresholds [El-Yaniv et al. '01, Lorenz et al. '09]

- Family of *k* thresholds: $\{\Phi_i\}_{1 \le i \le k}$

$$\frac{ALG}{OPT} = \frac{kU}{k\Phi_1} = \frac{\Phi_1 + (k-1)U}{k\Phi_2} + \frac{\Phi_1 + \Phi_2 + (k-2)U}{k\Phi_3} + \ldots + \frac{\sum_{i=1}^{k} \Phi_i}{kL} = \alpha$$

Note that: $\frac{kU}{k\Phi_1} = \alpha$ Also, we have that: $\frac{\sum_{i=1}^{k} \Phi_i}{kL} = \frac{\sum_{i=1}^{k} \Phi_i}{k\Phi_{i+1}} = \alpha$

α obtained by solving: $\frac{1 - 1/\theta}{1 - 1/\alpha} = \left(1 + \frac{1}{k\alpha}\right)^k$

Re-deriving k-min thresholds

- Idea: leverage balancing rule with knowledge of two "cases" for switching cost.

- Maintain two sets of thresholds: $\{\ell_i\}_{1 \leq i \leq k}$ $\{u_i\}_{1 \leq i \leq k}$ s.t. $\ell_i \leq u_i \ \forall i$

If we switch a lot, we should be more selective

$$\frac{ALG}{OPT} = \frac{\sum_{i=1}^{k} \ell_i + k2\beta}{kL + 2\beta} = \frac{\sum_{i=1}^{k} u_i + 2\beta}{kL + 2\beta} = \alpha$$

If we don't switch a lot, we can afford to accept larger cost

Re-deriving k-min thresholds (cont.)

- **Idea:** leverage balancing rule with two “cases” for switching cost.

$$\frac{ALG}{OPT} = \frac{\sum_{i=1}^{k} \ell_i + k2\beta}{kL + 2\beta} = \frac{\sum_{i=1}^{k} u_i + 2\beta}{kL + 2\beta} = \alpha$$

$$\frac{ALG}{OPT} = \frac{kU + 2\beta}{k\ell_1 + 2\beta} = \frac{\ell_1 + (k-1)U + 4\beta}{k\ell_2 + 2\beta} = \frac{u_1 + (k-1)U + 2\beta}{k\ell_2 + 2\beta} = \dots$$

$$\dots = \frac{\ell_1 + \ell_2 + (k-2)U + 6\beta}{k\ell_3 + 2\beta} = \frac{\ell_1 + u_2 + (k-2)U + 4\beta}{k\ell_3 + 2\beta} = \frac{u_1 + u_2 + (k-2)U + 2\beta}{k\ell_3 + 2\beta} = \dots = \alpha$$

Re-deriving k-min thresholds (cont.)

$$\frac{ALG}{OPT} = \frac{kU + 2\beta}{k\ell_1 + 2\beta} = \frac{\ell_1 + (k-1)U + 4\beta}{k\ell_2 + 2\beta} = \frac{u_1 + (k-1)U + 2\beta}{k\ell_2 + 2\beta} = \dots$$

$$\dots = \frac{\ell_1 + \ell_2 + (k-2)U + 6\beta}{k\ell_3 + 2\beta} = \frac{\ell_1 + u_2 + (k-2)U + 4\beta}{k\ell_3 + 2\beta} = \frac{u_1 + u_2 + (k-2)U + 2\beta}{k\ell_3 + 2\beta} = \dots = \alpha$$

- **Observation:** $u_i = \ell_i + 2\beta \;\; \forall 1 \leq i \leq k$ satisfies each potential "outcome".
  - Reduces the problem to solving for the family $\{\ell_i\}_{1 \leq i \leq k}$, which follows by using the algebraic techniques in [El-Yaniv et al. '01, Lorenz et al. '09].
  - What about α?

Redefining α with switching cost

- Existing α derivation [El-Yaniv et al. '01, Lorenz et al. '09]

α obtained by solving: $$\frac{1 - 1/\theta}{1 - 1/\alpha} = \left(1 + \frac{1}{k\alpha}\right)^k$$ where $\theta = U/L$

algebraic manipulation of the expression for Φ_{k+1}, which is itself obtained by comparing adjacent terms from balancing rule

$$\Phi_{k+1} = L = U\left[1 - \left(1 - \frac{1}{\alpha}\right)\left(1 + \frac{1}{k\alpha}\right)^k\right]$$

- To incorporate switching cost, derive expression for individual thresholds $\{\ell_i\}_{1 \le i \le k}$.

"θ" should depend on $U, L, \& \beta$.

Deriving for individual thresholds?

- Solving balancing rule for $\{\ell_i\}_{1\leq i\leq k}$

$$\frac{ALG}{OPT} = \frac{kU + 2\beta}{k\ell_1 + 2\beta} = \frac{\ell_1 + (k-1)U + 4\beta}{k\ell_2 + 2\beta} = \frac{\ell_1 + \ell_2 + (k-2)U + 6\beta}{k\ell_3 + 2\beta} = \alpha$$

$$\ell_1 = \frac{U}{\alpha} + 2\beta\left(\frac{1}{\alpha k} - \frac{1}{k}\right)$$

$$\ell_2 = \frac{\frac{U}{\alpha} + 2\beta\left(\frac{1}{\alpha k} - \frac{1}{k}\right)}{\alpha k} + \frac{U}{\alpha} - \frac{U}{\alpha k} + \frac{2\beta}{\alpha k} + 2\beta\left(\frac{1}{\alpha k} - \frac{1}{k}\right)$$

$$\ell_3 = \frac{\frac{U}{\alpha} + 2\beta\left(\frac{1}{\alpha k} - \frac{1}{k}\right)}{\alpha k} + \frac{\frac{\frac{U}{\alpha} + 2\beta\left(\frac{1}{\alpha k} - \frac{1}{k}\right)}{\alpha k} + \frac{U}{\alpha} - \frac{U}{\alpha k} + \frac{2\beta}{\alpha k} + 2\beta\left(\frac{1}{\alpha k} - \frac{1}{k}\right)}{\alpha k} + \frac{U}{\alpha} - \frac{U}{\alpha k} + \frac{4\beta}{\alpha k} + 2\beta\left(\frac{1}{\alpha k} - \frac{1}{k}\right)$$

Deriving for individual thresholds?

- Solving balancing rule for $\{\ell_i\}_{1\le i\le k}$

$$\ell_i = U\left[1-\left(1-\frac{1}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{i-1}\right] + 2\beta\left[\boxed{\sum_{j=1}^{i-1}\frac{(i-j)}{\alpha^j k^j}} - \frac{1}{k}\left(1-\frac{1}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{i-1}\right]$$

$$i = 1 \rightarrow \ell_1 = \ldots + 0$$

$$i = 2 \rightarrow \ell_2 = \ldots + \frac{2\beta}{\alpha k}$$

$$i = 3 \rightarrow \ell_3 = \ldots + \frac{2\beta}{\alpha^2 k^2} + \frac{4\beta}{\alpha k}$$

$$i = 3 \rightarrow \ell_4 = \ldots + \frac{2\beta}{\alpha^3 k^3} + \frac{4\beta}{\alpha^2 k^2} + \frac{6\beta}{\alpha k}$$

not a factorable sequence?

Deriving for individual thresholds?

- Solving balancing rule for $\{\ell_i\}_{1\le i\le k}$

$$\ell_i = U\left[1-\left(1-\frac{1}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{i-1}\right]+2\beta\left[\sum_{j=1}^{i-1}\frac{(i-j)}{\alpha^j k^j}-\frac{1}{k}\left(1-\frac{1}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{i-1}\right]$$

alternate form: $$\ell_i = U+2\beta\sum_{j=1}^{i-1}\frac{(i-j)}{\alpha^j k^j}-\left(U+\frac{2\beta}{k}\right)\left(1-\frac{1}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{i-1}$$

- Solve for α?

$$\ell_{k+1} = L = U+2\beta\sum_{j=1}^{k}\frac{(i-j)}{\alpha^j k^j}-\left(U+\frac{2\beta}{k}\right)\left(1-\frac{1}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{k}$$

$$U-L+2\beta\sum_{j=1}^{k}\frac{(i-j)}{\alpha^j k^j}=\left(U+\frac{2\beta}{k}\right)\left(1-\frac{1}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{k}$$

Deriving for individual thresholds?

- Observation: balancing rule for $\{u_i\}_{1\le i\le k}$

$$\frac{u_1+(k-1)U+2\beta}{k\ell_2+2\beta}=\frac{u_1+(k-1)U+2\beta}{k(u_2-2\beta)+2\beta}$$

$$u_i = U\left[1-\left(1-\frac{1}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{i-1}\right]+2\beta\left[\left(\frac{1}{k\alpha}-\frac{1}{k}+1\right)\left(1+\frac{1}{k\alpha}\right)^{i-1}\right]$$

- Solve for α?

$$u_{k+1}=L+2\beta=U-\left(U+\frac{U}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{k}+\left(2\beta-\frac{2\beta}{k}+\frac{2\beta}{k\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{k}$$

$$U-L-2\beta=\left(U+\frac{U}{\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{k}-\left(2\beta-\frac{2\beta}{k}+\frac{2\beta}{k\alpha}\right)\left(1+\frac{1}{k\alpha}\right)^{k}$$

$$\frac{U-L-2\beta}{\left(U+\frac{U}{\alpha}\right)-\left(2\beta-\frac{2\beta}{k}+\frac{2\beta}{k\alpha}\right)}=\left(1+\frac{1}{k\alpha}\right)^{k}$$