

Randomized binary and tree search under pressure



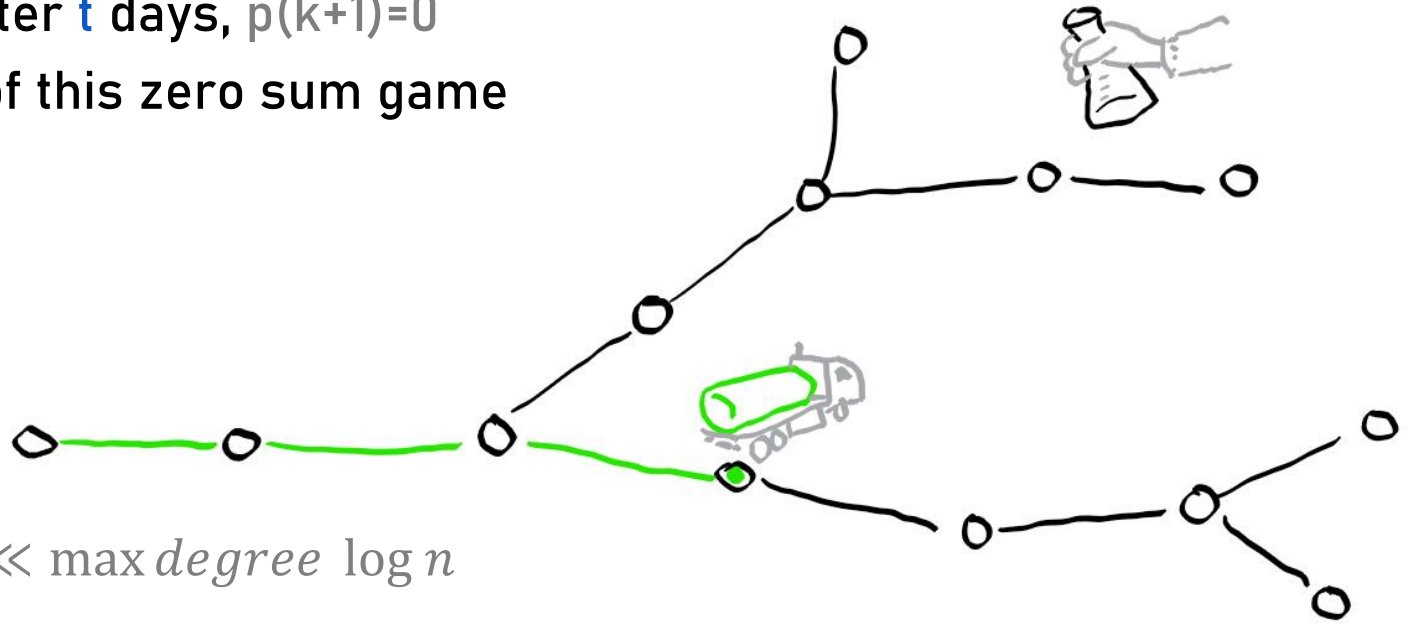
Agustín Caracci
Christoph Dürr
José Verschae



June 2024, Séminaire Parisien d'Optimisation

Motivation

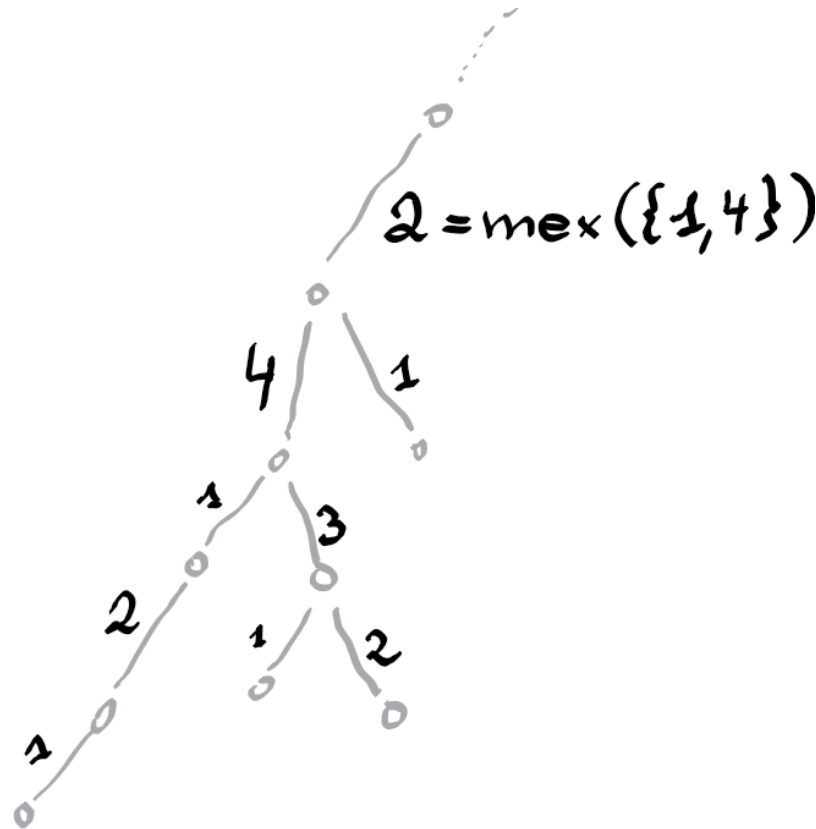
- River is a tree on n nodes
- Pollution is happening during k days at some node (**hider**)
- Inspection is done one edge per day (**searcher**)
- Profit $p(t)$ if hider is located after t days, $p(k+1)=0$
- Find mixed Nash equilibrium of this zero sum game



- Assumption: $\max \text{degree} \leq k \ll \max \text{degree} \log n$

Related work

- Fernando Cicalese studied many variants
- [Lam, Yue, 2001] linear time computing a strategy to locate hider with minimum number of edge queries in a tree
- was known for vertex queries, Schäffer 1989
- Locating hider in a *graph* is hard (under the strongly exponential time hypothesis)

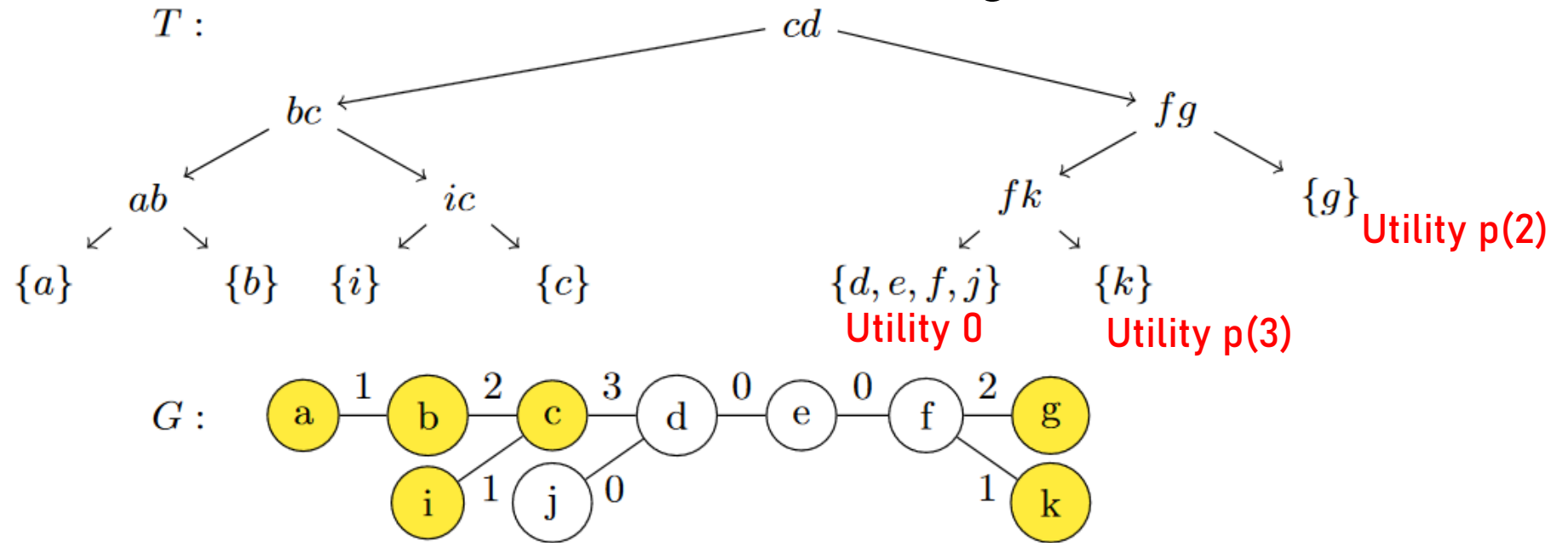


- Search strategy = edge labels
 - Label represents remaining number of queries
 - Query highest label edge
 - Continue in the indicated connected component
- To compute search strategy
 - Orient tree arbitrarily
 - Compute bottom up labels
 - Visibility set of edge v contains labels below not hidden by higher labels
 - Set label of v to be the minimum excluded number of set

The game

- Hider selects a distribution on vertices of the tree **G**
- **Searcher** selects a distribution on deterministic search strategies **T**

k=3 queries



Our contributions

Given a distribution of the hider on the vertices, find a deterministic strategy which locates the hider with a maximum probability

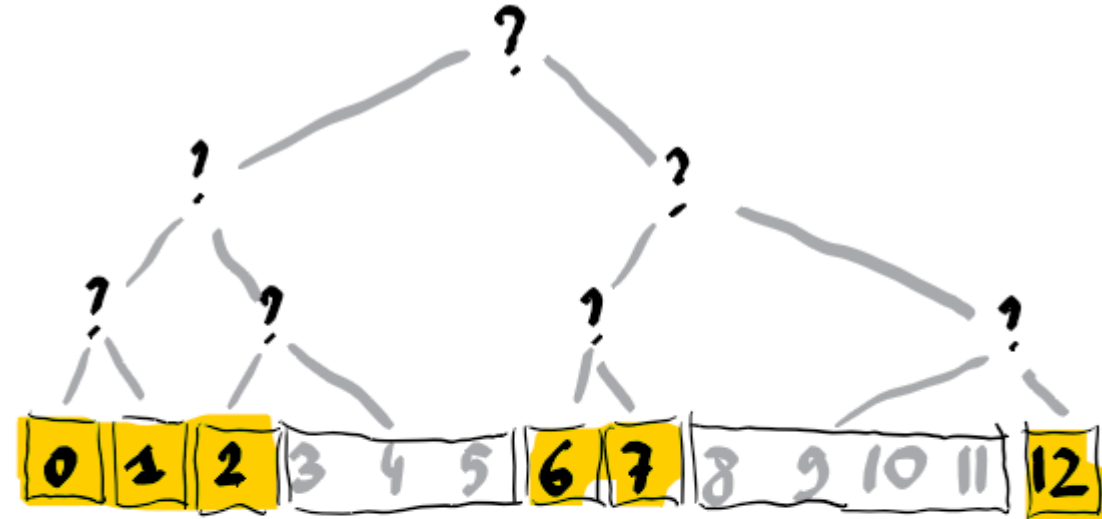
Use a separation oracle on the linear programming formulation of the game



model	Compute searcher's best response	Compute Nash equilibrium
Unit profit ($p(t) = 1$ if $t \leq k$, 0 else) Game on a path	In time $O(2^k n)$	In time $O(\log n)$
General profit Game on a tree	In time $O(2^{2k} n^2)$ NP-hard for $k=n-1$	In time $2^{O(k)} \text{poly}(n)$

Unit profit, game on a path

- A deterministic strategy partitions the path $\{0, \dots, n - 1\}$ into 2^k parts
- Only singleton parts locate the hider
- Hence **efficient strategies** correspond to intervals of size $c := 2^k - 2$
- or of size $c + 1$ if the *modular interval* includes the border



0 1 2 3 4 5 6 7 8 9 10 11 12

0 1 2 3 4 5 6 7 8 9 10 11 12

Rough value of the game

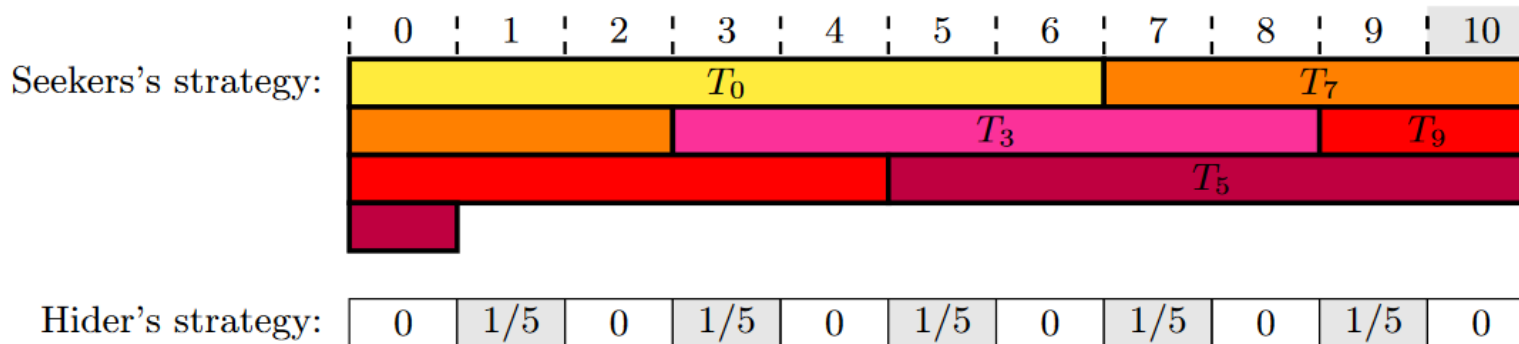
- If rivers were **circular**, an efficient strategy would cover c/n fractions of the vertices
- Uniform random choice of an efficient strategies leads to a game value of c/n
- But on a **path** the border matters, we distinguish two cases...



Case $\gcd(n-1, c) > 1$: searcher

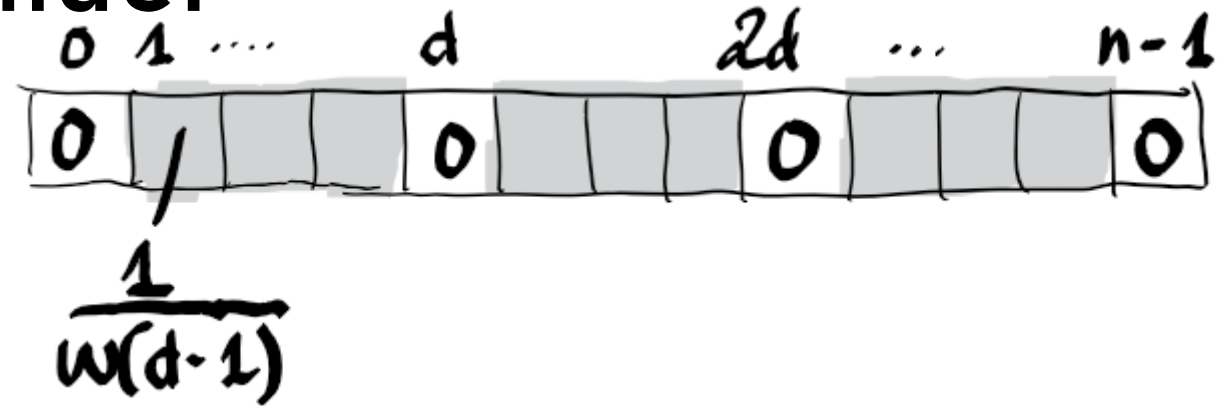
- example $n=11, k=3, c=6$

- The searcher chooses uniformly among a collection of 5 efficient strategies
- The corresponding intervals round-Robin **cover** the n columns of some matrix
- Ignoring the last column, first interval T_0 has size $c+1$, all others have size c
- First column is over covered,
By shifting the intervals, searcher could over cover the columns 0,2,4,6,8 or 10,
Hence hider avoids those columns
- Value of the game is $\geq 3/5$



- **Cover a matrix with $n-1$ columns**
- Place efficient strategies one after another, until last interval ends on column 0 (in case $\gcd(n-1, c) > 1$) or on column $n-1$ (in case $\gcd(n-1, c) = 1$)
- w = number of used strategies
- h = number of times every column (> 0) is covered
- $w = (n-1)/d, h = c/d$ for $d = \gcd(n-1, c)$
- $0 < w \leq n-2$ is minimal such that $h(n-1) - wc = 0$
- $w = 10/2 = 5, h = 6/2 = 3, d = \gcd(10, 6) = 2$

Case $\gcd(n-1, c) > 1$: hider



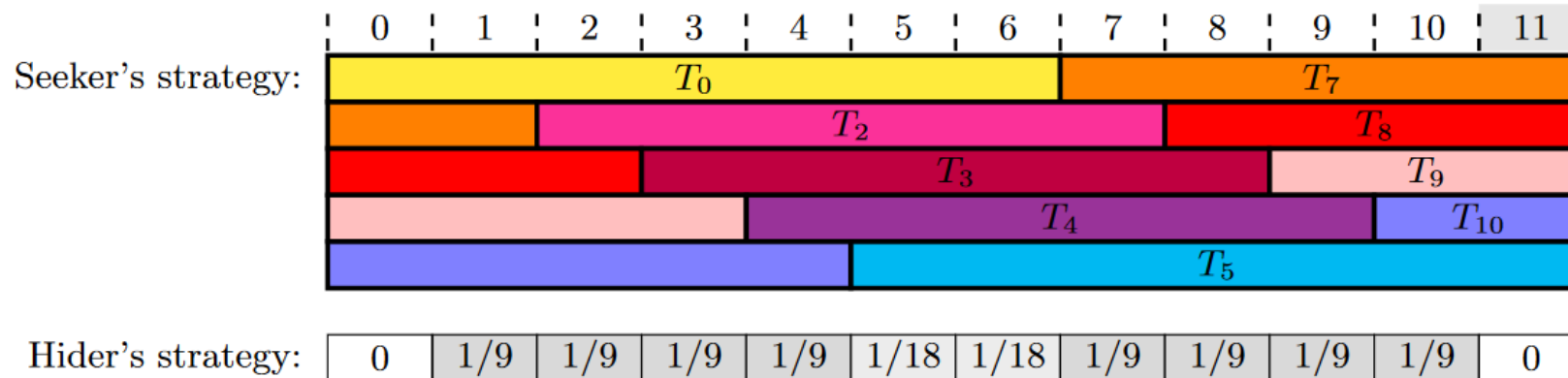
- The hider chooses probability $1/w(d-1)$ for each column v which is **not** a multiple of d
- **Sum of probabilities is 1**: there are $(n-1)/d = w$ multiples of d in $\{0, \dots, n-2\}$. Hence total probability mass is

$$\frac{n-1-w}{w(d-1)} = \frac{d(n-1) - (n-1)}{wd(d-1)} = \frac{n-1}{wd} = 1$$

- **Every efficient strategy** captures a probability mass of $\frac{c - \frac{c}{d}}{w(d-1)} = \frac{c(d-1)}{(n-1)(d-1)} = \frac{c}{n-1}$
- Every strategy captures at most $\frac{c}{n-1}$
- Value of the game is $\frac{c}{n-1} = \frac{h}{w}$

Case $\gcd(n-1, c)=1$: searcher

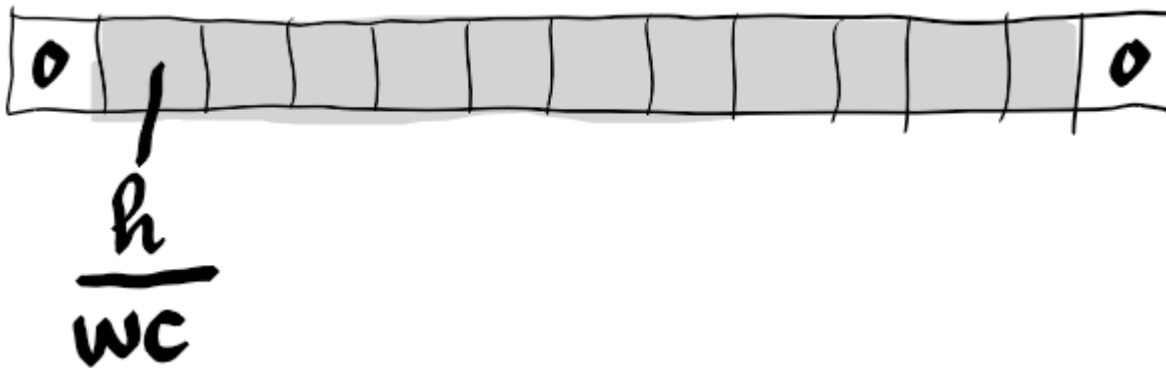
- example $n=12, k=3, c=6$
- Value of the game is $\frac{h}{w} = \frac{5}{9}$
- w = number of intervals (deterministic strategies)
 h = height of this matrix
- $0 < w \leq n - 2$ is minimal such that $h(n-1) - wc = 1$
- Borders of intervals partition the path,
 Hider chooses uniformly among those parts and then uniformly within, avoiding the border of the path



Case $\gcd(n-1, c)=1$: suboptimal hider

- Distribute h/wc on columns $[1, n-2]$
- Probability mass captured by efficient strategies is $ch/wc=h/w$ (perfect)
- But total probability mass is (not enough)

$$\frac{h(n-2)}{wc} = \frac{h(n-2)}{h(n-1)-1} < 1$$

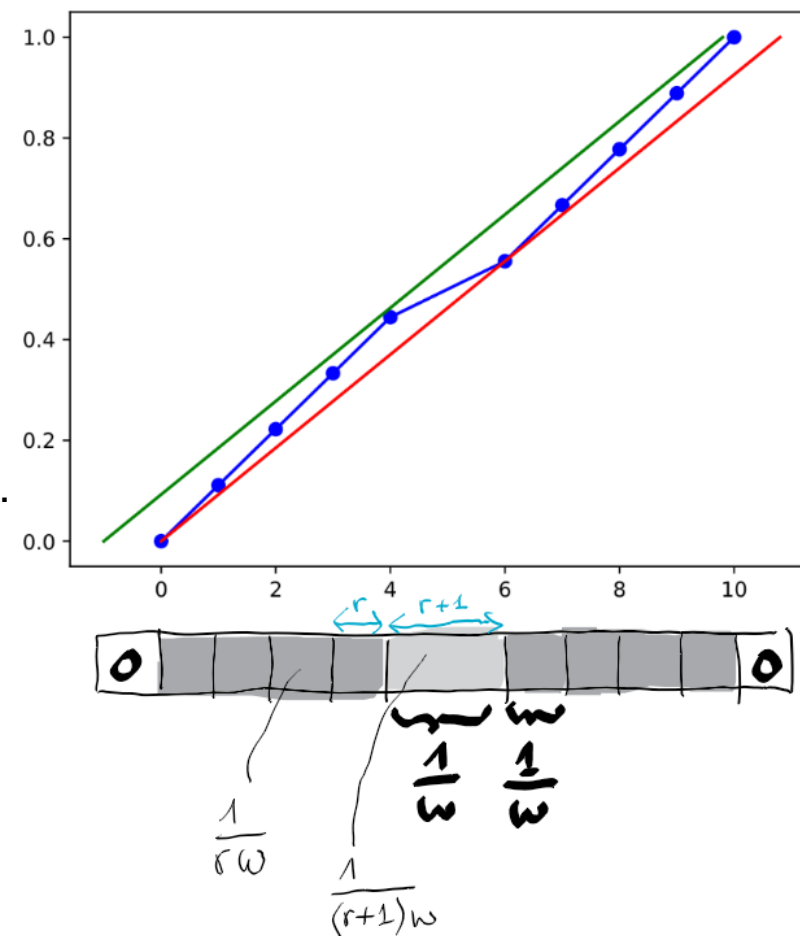


Case $\gcd(n-1, c)=1$: optimal hider

- Choose distribution y on columns $[1, n-2]$
- Define ideal probability mass in $[1, v]$ as $g(v) := v \frac{h}{wc}$
- Hider wants

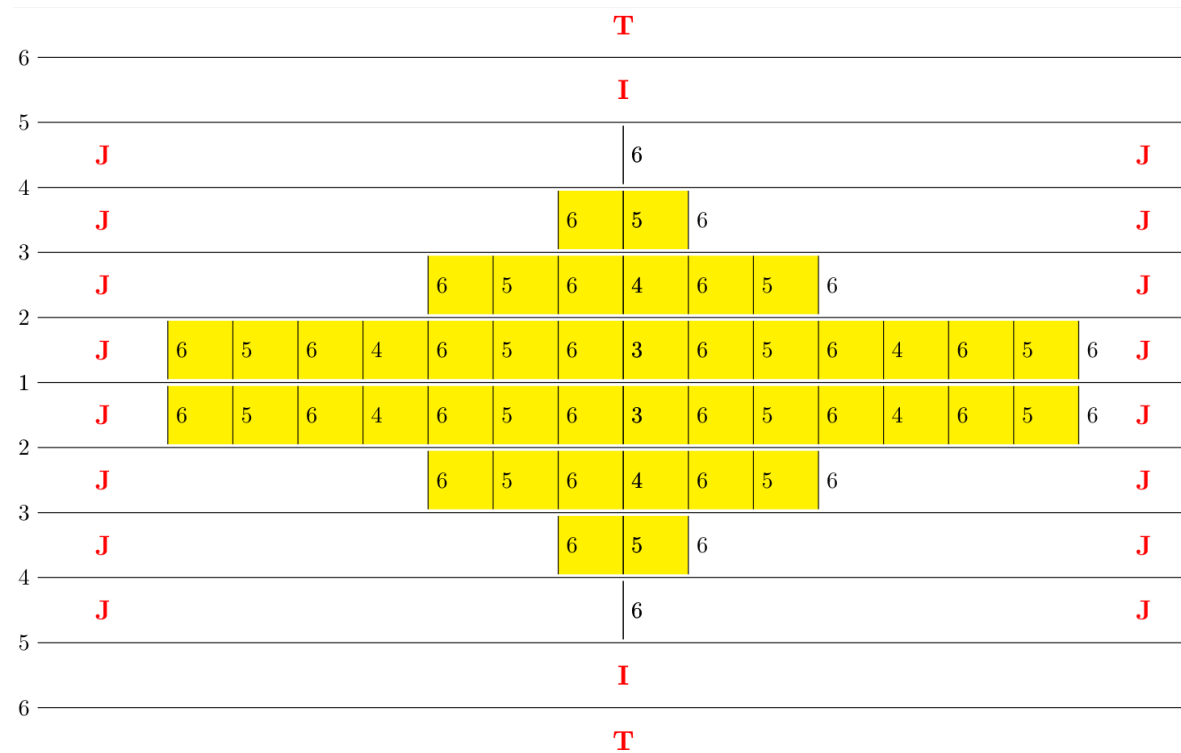
$$g(v) \leq y([1, v]) < g(v+1)$$

- Partition** $[1, n-2]$ into w intervals of length r or $r+1$ for $r = \lfloor c/h \rfloor$
Formally: start at $v=1$.
Choose largest $r^* \in \{r, r+1\}$ such that $g(v+r^*-1) \leq y([1, v-1]) + 1/w$.
Distribute uniformly $1/w$ on next interval $[v, v+r^*-1]$.
Update $v = v+r^*$.
- Periodicity**: $y_v = y_{v+c}$. Hence every efficient strategy captures $y([1, c]) = h/w$
- Value of the game is h/w



Coming up

- Searching with guillotine cuts in a grid



Thank you, space for questions