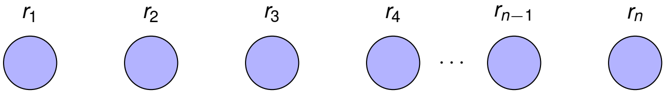
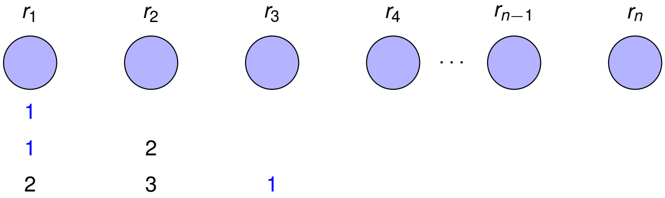


Elements arrive in uniform random order.



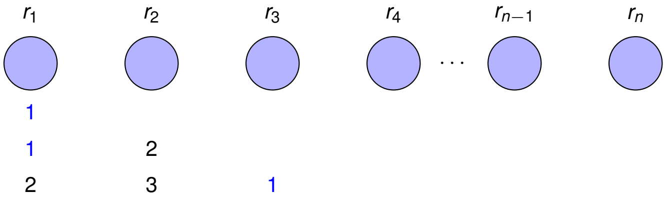
Elements arrive in uniform random order.

Total order $\succ$ over R.



Elements arrive in uniform random order.

Total order $\succ$ over R.



Select one element: What is the best stopping rule?

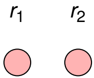
Strong algorithms for the Ordinal Matroid Secretary Problem

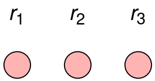
José Soto — UChile
Abner Turkieltaub — UChile
Victor Verdugo — UChile-ENS

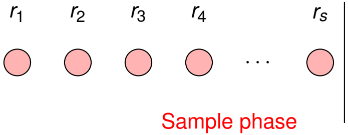
Approximation and Networks, Collége de France

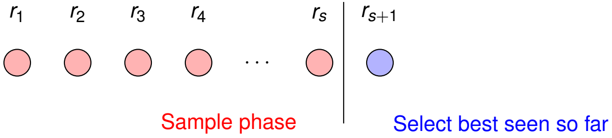
Paris, June 7, 2018

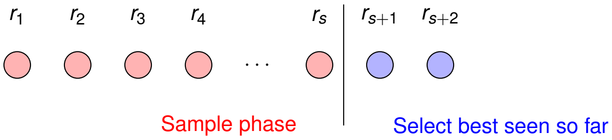

r1

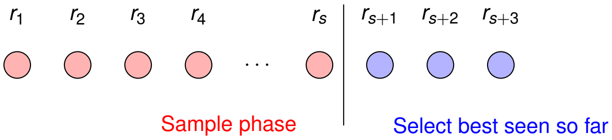

r_1
r_2

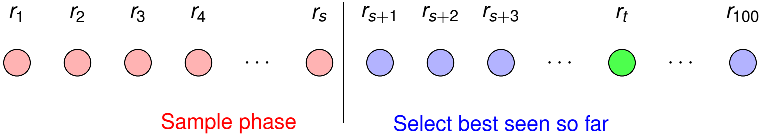

r_1
r_2
r_3

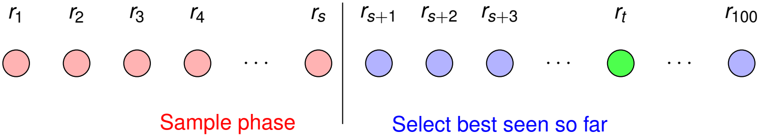

r_1
r_2
r_3
r_4
r_s
Sample phase


r_1
r_2
r_3
r_4
r_s
r_{s+1}
Sample phase
Select best seen so far

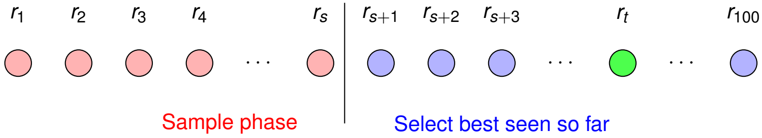

r_1
r_2
r_3
r_4
$\cdots$
r_s
r_{s+1}
r_{s+2}
Sample phase
Select best seen so far


r_1
r_2
r_3
r_4
r_s
r_{s+1}
r_{s+2}
r_{s+3}
Sample phase
Select best seen so far


r_1
r_2
r_3
r_4
r_s
r_{s+1}
r_{s+2}
r_{s+3}
r_t
r_{100}
Sample phase
Select best seen so far



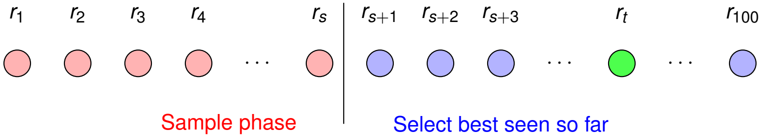
$$P(\text{select best}) \approx -\frac{s}{100}\ln\left(\frac{s}{100}\right).$$



$$\mathrm{P}(\text{select best}) \approx -\tfrac{s}{100} \ln\left(\tfrac{s}{100}\right).$$

($\star$) Conditioning on the history at time t,

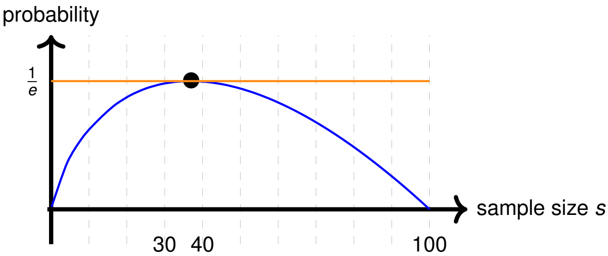
$$\mathrm{P}_t(\text{select best}) \geq \prod_{j=s+1}^{t-1} \mathrm{P}_t(r_j \text{ is not the second best}) = \prod_{j=s+1}^{t-1} \frac{j-1}{j} = \frac{s}{t-1},$$



($\star$) Conditioning on the history at time t,

$$\mathrm{P}_t(\text{select best}) \geq \prod_{j=s+1}^{t-1} \mathrm{P}_t(r_j \text{ is not the second best}) = \prod_{j=s+1}^{t-1} \frac{j-1}{j} = \frac{s}{t-1},$$

$$\mathrm{P}(\text{select best}) \geq \frac{s}{100} \sum_{t=s+1}^{100} \frac{1}{t-1} \approx \frac{s}{100} \int_s^{100} \frac{dx}{x} = -\frac{s}{100} \ln\left(\frac{s}{100}\right)$$

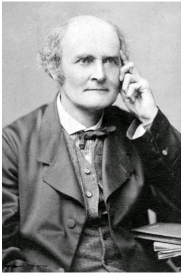


$s = 37 \approx 100/e$ maximizes the probability!

Who solved it?

Lindley 1961, scientific publication.

Scientific American 1960, puzzle.



For every element in OPT, P(element is selected) $\geq 1/\alpha$.

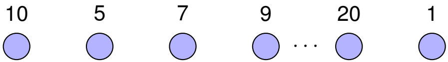
Obs. By picking $s \sim$ Binomial$(100, 1/e)$, algorithm is $1/e$ prob-competitive.

$$E_s\left(\frac{1}{100}\sum_{j=s+1}^{100}\prod_{j=s+1}^{t-1}\frac{j-1}{j}\right) \geq -p\ln p \text{ ... optimize here}$$

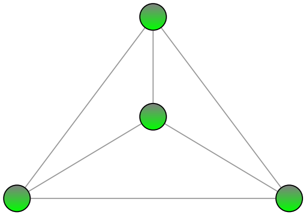
For every element in OPT, P(element is selected) $\geq 1/\alpha$.

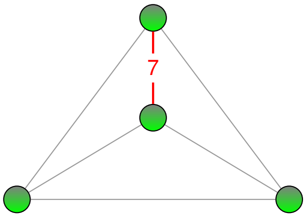
Obs. By picking $s \sim \text{Binomial}(100, 1/e)$, algorithm is $1/e$ prob-competitive.

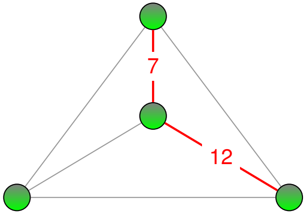
$$E_s\left(\frac{1}{100}\sum_{j=s+1}^{100}\prod_{j=s+1}^{t-1}\frac{j-1}{j}\right) \geq -p\ln p \text{ ... optimize here}$$

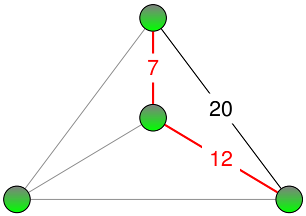


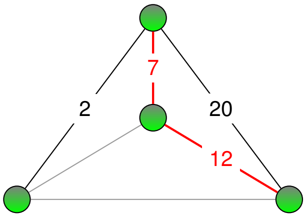
Utility competitive $\sim \mathbb{E}(w(ALG)) \geq \frac{1}{\alpha}w(OPT)$.

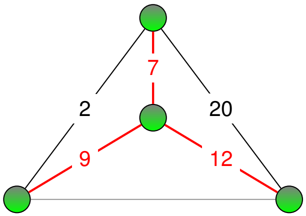


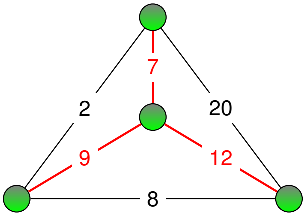

7

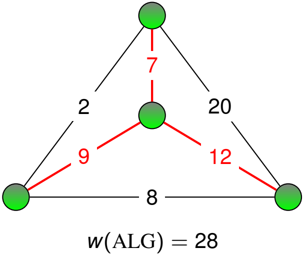

7
12


7
20
12


7
2
20
12


7
2
20
9
12

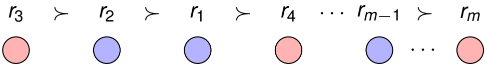

7
2
20
9
12
8



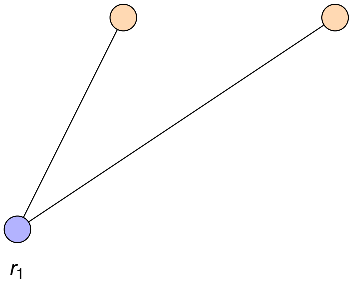
This solution is $41/28 \approx 1.46$ utility-competitive.

Offline: Select greedily

1. Greedy in decreasing order returns the optimal $\mathrm{OPT}(E)$.
2. For every $F \subseteq E$, $\mathrm{OPT}(E) \cap F \subseteq \mathrm{OPT}(F)$.

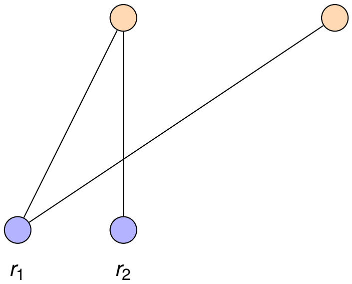


OPT



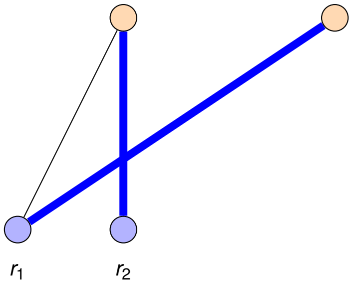
neighbours

terminals arrive online!



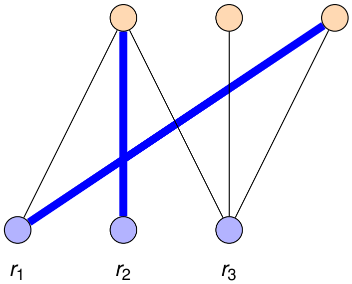
neighbours

terminals arrive online!



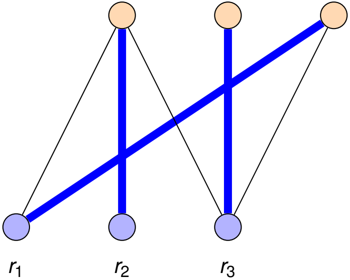
neighbours

terminals arrive online!



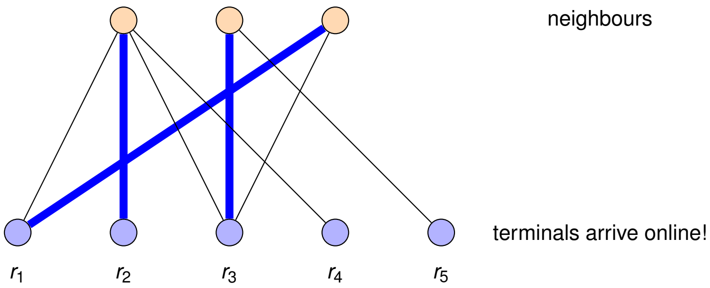
neighbours

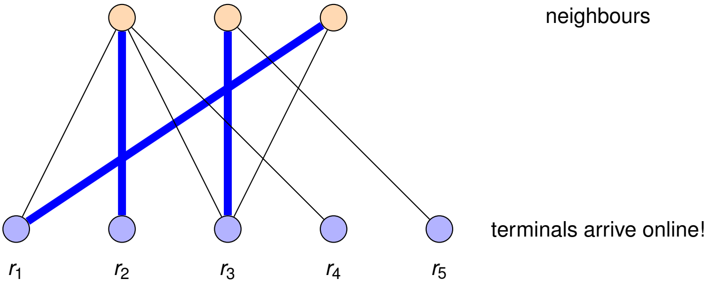
terminals arrive online!



neighbours

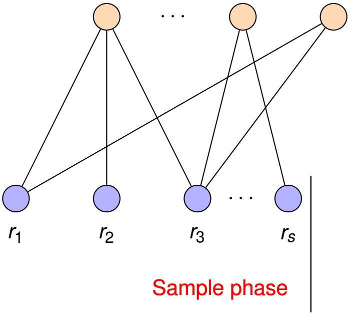
terminals arrive online!


neighbours
terminals arrive online!
r_1
r_2
r_3
r_4
r_5

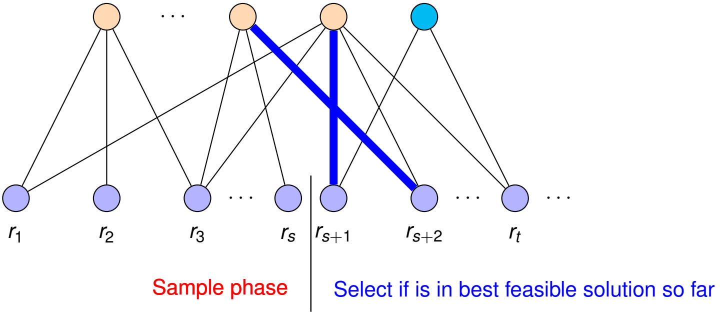


$\{r_1, r_2, r_3\}$ is a feasible selection

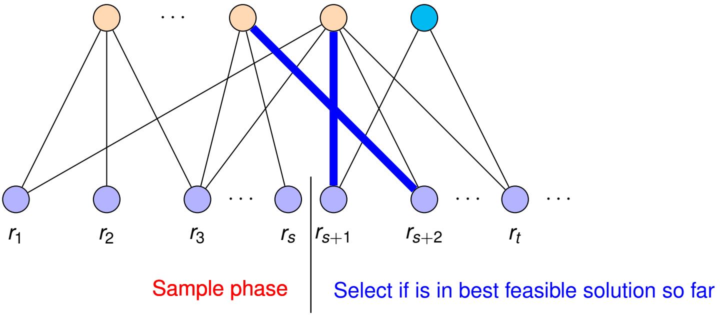
$\{r_2, r_3, r_4\}$ is **not** a feasible selection



If $r_t \in \mathrm{OPT}$, it could only be blocked by **one** other current best!



If $r_t \in \mathrm{OPT}$, it could only be blocked by **one** other current best!



If $r_t \in \mathrm{OPT}$, it could only be blocked by **one** other current best!

$$\mathrm{P}_t(\text{select } r_t) \geq \prod_{j=s+1}^{t-1} \mathrm{P}_t(r_j \in OPT_j \text{ is not matched to green}) = \prod_{j=s+1}^{t-1} \frac{j-1}{j} = \frac{s}{t-1}.$$

More generally ...

Suppose we have an algorithm for a matroid class such that:

(Feasibility) Returns an independent set.

(Sampling) Sample $s \sim \text{Bin}(m, p)$ elements and reject them.

(k-forbidden) Let r an element in OPT arriving at time $t > s$.

For each $s < i < t$, a random set F_i of size atmost k, such that:
If for each $s < i < t$, the element arriving $\notin F_i$ then r is selected.

More generally ...

Suppose we have an algorithm for a matroid class such that:

(Feasibility) Returns an independent set.

(Sampling) Sample $s \sim \text{Bin}(m, p)$ elements and reject them.

(k-forbidden) Let r an element in OPT arriving at time $t > s$.

For each $s < i < t$, a random set F_i of size atmost k, such that:
If for each $s < i < t$, the element arriving $\notin F_i$ then r is selected.

Thm. By setting the right sampling probability, a k-forbidden algorithm is α prob-competitive, where

$$\alpha = \begin{cases} e & \text{if } k = 1, \\ k^{k/(k-1)} & \text{if } k \geq 2. \end{cases}$$

More generally ...

Suppose we have an algorithm for a matroid class such that:

(Feasibility) Returns an independent set.

(Sampling) Sample $s \sim \text{Bin}(m, p)$ elements and reject them.

(k-forbidden) Let r an element in OPT arriving at time $t > s$.
For each $s < i < t$, a random set F_i of size atmost k, such that:
If for each $s < i < t$, the element arriving $\notin F_i$ then r is selected.

Thm. By setting the right sampling probability, a k-forbidden algorithm is α prob-competitive, where

$$\alpha = \begin{cases} e & \text{if } k = 1, \\ k^{k/(k-1)} & \text{if } k \geq 2. \end{cases}$$

Proof. Study the quantity

$$E_s \left(\frac{1}{100} \sum_{j=s+1}^{100} \prod_{j=s+1}^{t-1} \frac{j-k}{j} \right).$$

Further applications of the technique

	Previous	Our results (p)
Transversal	e (u)	e
μ-Gammoid	μe (u)	$\mu^{\mu/(\mu-1)}$
Graphic	$2e$ (u)	4
Laminar	9.6 (p)	$3\sqrt{3} \approx 5.19$
k-column sparse	ke (u)	$k^{k/(k-1)}$
Matching	-	4
Graph packings	-	$\mu^{\mu/(\mu-1)}$
k-framed	-	$k^{k/(k-1)}$
Hypergraphic	-	4
Semiplanar	-	$4^{4/3}$

Good necessary condition for optimality

e.g. current offline optimum

$+$

Small forbidden sets

e.g. study some combinatorial witness

- $O(1)$-forbidden algorithm for general matroids.
- **Strongest version:** 1-forbidden algorithm for general matroids.
- Apply the framework over non-matroidal settings.

- $O(1)$-forbidden algorithm for general matroids.
- **Strongest version:** 1-forbidden algorithm for general matroids.
- Apply the framework over non-matroidal settings.

Merci!