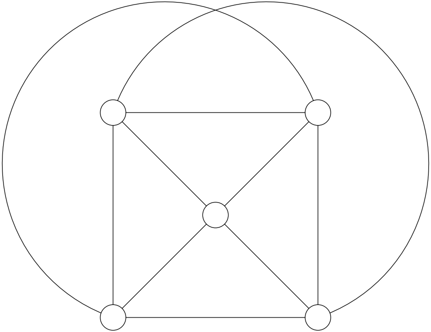
Principal Partition and the Matroid Secretary Problem

José A. Soto

Department of Mathematics
M.I.T.

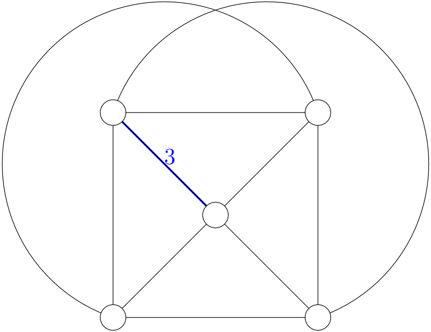
SIAM Conference on Optimization (OP11)
May 17, 2011.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



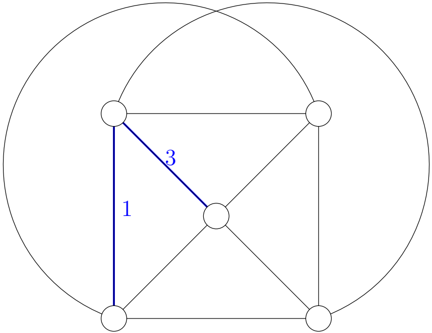
- Given a matroid.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



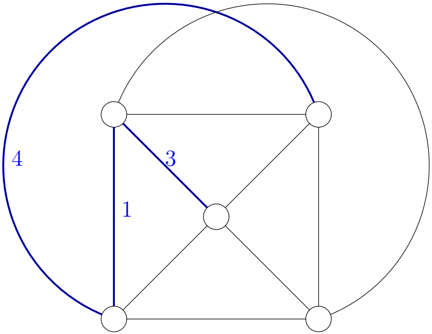
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



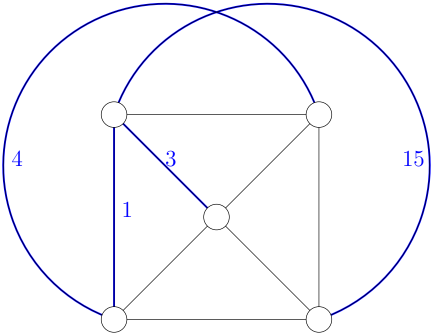
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



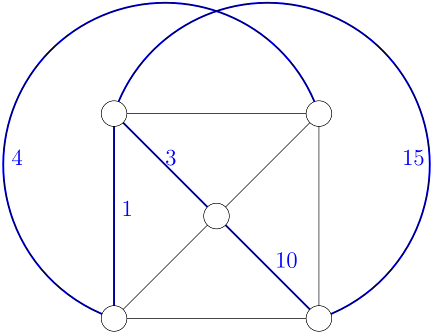
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



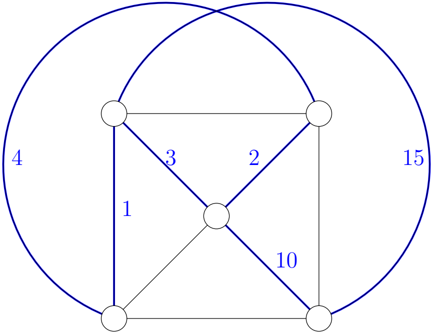
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



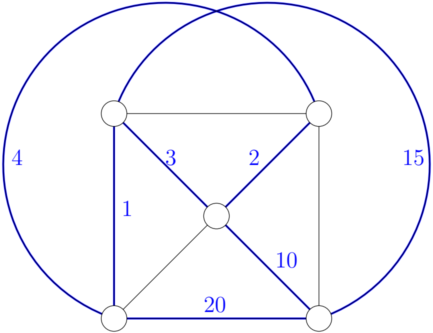
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



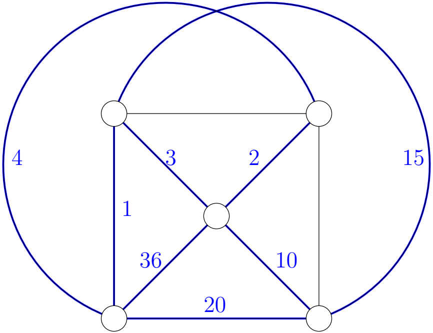
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



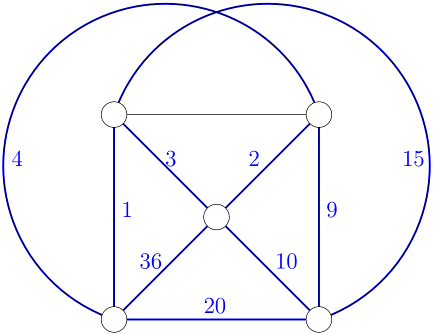
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



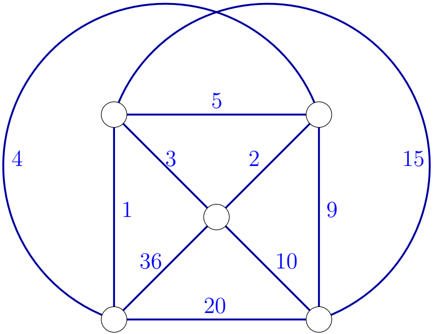
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



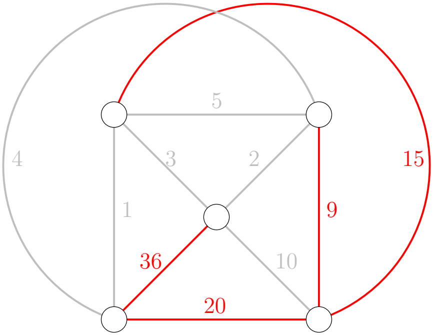
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



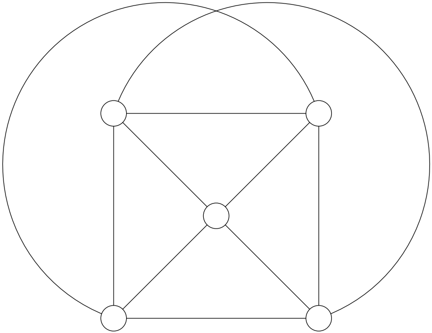
- Given a matroid.
- Elements' weights are revealed in certain (random) order.

MSP: Introduction [Babaioff, Immorlica, Kleinberg]



- Given a matroid.
- Elements' weights are revealed in certain (random) order.
- Want to select independent set of high weight.
  (In online way / secretary problem setting)

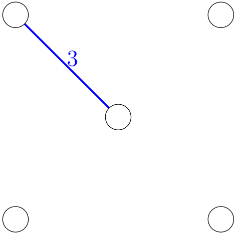
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

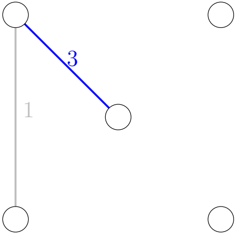
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

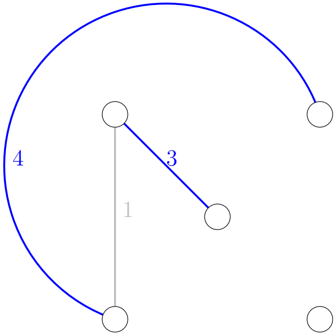
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

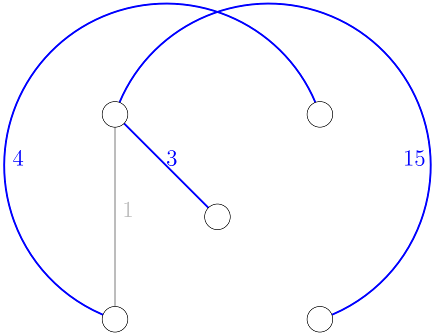
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

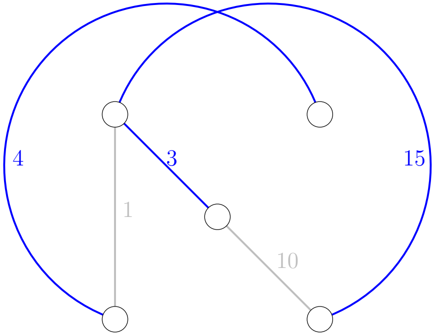
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

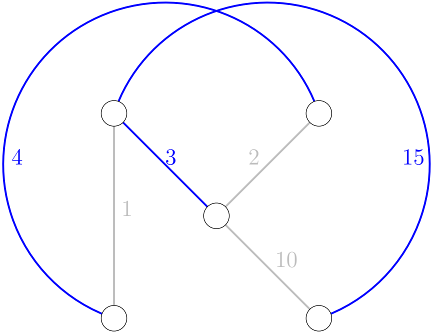
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

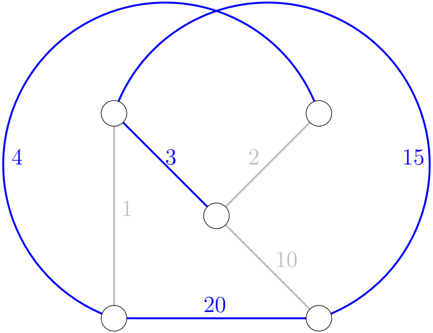
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

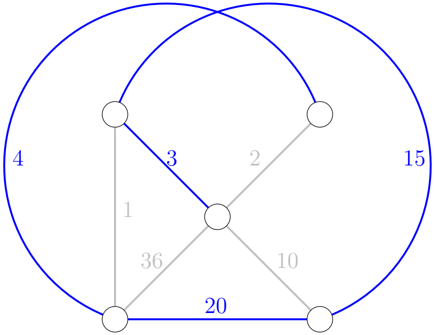
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

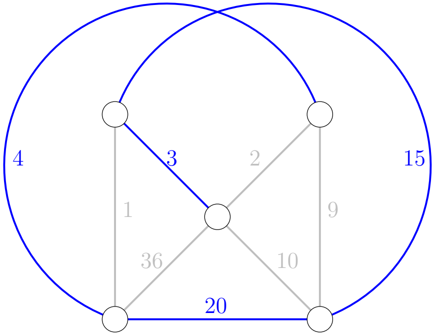
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

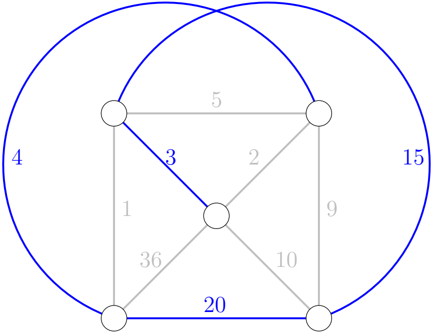
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

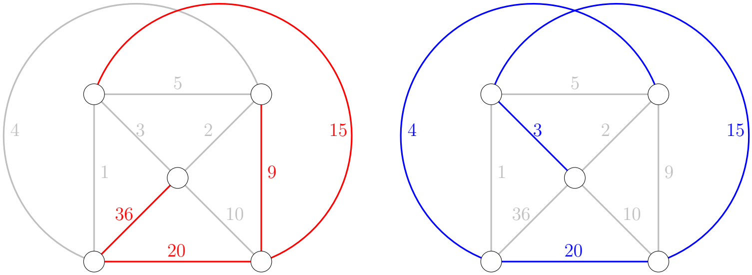
MSP: Introduction (II)



Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

MSP: Introduction (II)



$w(\text{OPT}) = 80$

$w(\text{ALG}) = 42$

Rules

- We accept or reject an element when its weight is revealed.
- Accepted elements must form an independent set.

Special Cases

Classical / Multiple choice

- Hire one person (or at most *r*).
- Sell one item to best bidder (or sell *r* identical items).

Models: Given a known matroid $\mathcal{M} = (E, \mathcal{I})$.

Opponent selects n **weights.**
$w_1 \geq w_2 \geq \cdots \geq w_n \geq 0$

then

The weights are assigned either:
adversarially or at **random.**

and *independently*

The presentation order is chosen:
adversarially or at **random.**

Models: Given a known matroid $\mathcal{M} = (E, \mathcal{I})$.

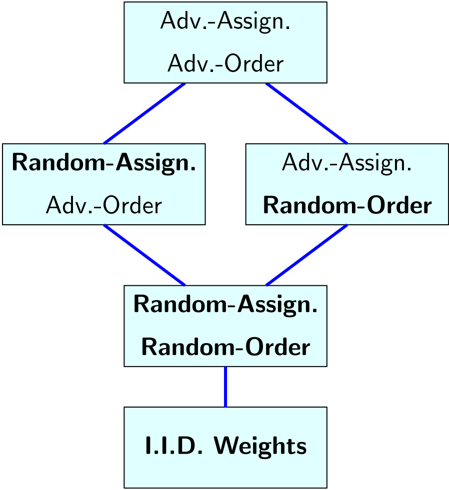
Opponent selects n **weights.**
$w_1 \geq w_2 \geq \cdots \geq w_n \geq 0$

then

The weights are assigned either:
adversarially or at **random.**

and *independently*

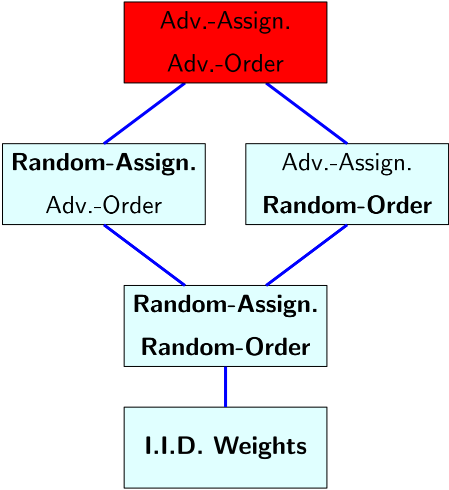
The presentation order is chosen:
adversarially or at **random.**



Models: Given a known matroid $\mathcal{M} = (E, \mathcal{I})$.

- (Adv.-Assign. Adv.-Order)
  Hard: n-competitive ratio

  [Babaioff, Immorlica, Kleinberg 07]
  Conjecture: O(1)-competitive algorithm for all other models.

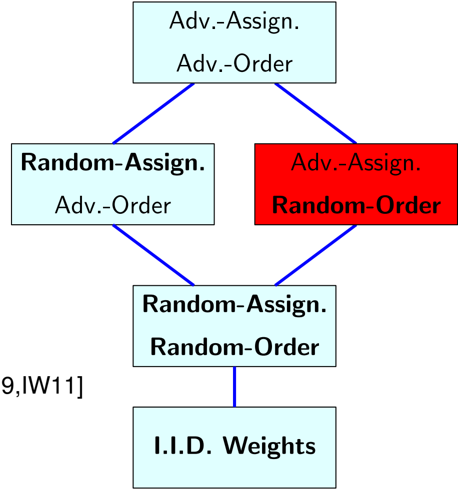


Models: Given a known matroid $\mathcal{M} = (E, \mathcal{I})$.

- (Adv.-Assign. Adv.-Order)
  Hard: n-competitive ratio

  [Babaioff, Immorlica, Kleinberg 07]
  Conjecture: O(1)-competitive algorithm for all other models.

- (Adv.-Assign. Random-Order)
  O(1) for partition, graphic, transversal, laminar.
  [L61,D63,K05,BIK07,DP08,KP09,BDGIT09,IW11]
  O$(\log \mathrm{rk}(\mathcal{M}))$ **for general matroids** [BIK07].



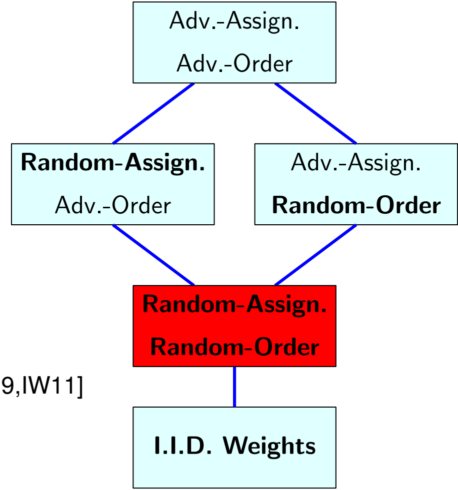
Models: Given a known matroid $\mathcal{M} = (E, \mathcal{I})$.

- (Adv.-Assign. Adv.-Order)
  Hard: n-competitive ratio

  [Babaioff, Immorlica, Kleinberg 07]
  Conjecture: $O(1)$-competitive algorithm for all other models.

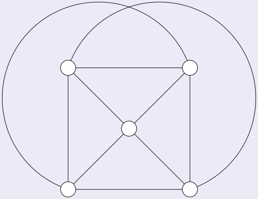
- (Adv.-Assign. Random-Order)
  $O(1)$ for partition, graphic, transversal, laminar.
  [L61,D63,K05,BIK07,DP08,KP09,BDGIT09,IW11]
  $O(\log \mathrm{rk}(\mathcal{M}))$ for general matroids [BIK07].

- (Random-Assign. Random-Order)
  [S11a] $O(1)$ for general matroids.



Random-Assignment Random-Order.

Data



Known Matroid

$\xleftarrow{\sigma}$ *rand.assign.*

$\xrightarrow{\pi}$ *rand.order*

$\mathbf{W}: \mathbf{w_1} \geq \mathbf{w_2} \geq \cdots \geq \mathbf{w_n} \geq \mathbf{0}.$

Hidden weight list

Algorithm
accepts of rejects.

Objective

Return an independent set $\mathrm{ALG} \in \mathcal{I}$ such that:

$$\mathbb{E}_{\pi,\sigma}[w(\mathrm{ALG})] \geq \Omega(1) \cdot \mathbb{E}_{\sigma}[w(\mathrm{OPT})], \text{ where}$$

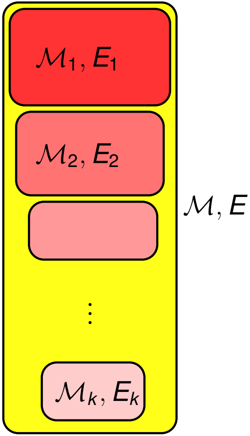
OPT is the optimum base of $\mathcal{M}$ under assignment σ. (Greedy)

Divide and Conquer to get $O(1)$-competitive algorithm.

For a general matroid $\mathcal{M} = (E, \mathcal{I})$:

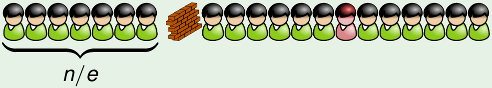
Find matroids $\mathcal{M}_i = (E_i, \mathcal{I}_i)$ with $E = \bigcup_{i=1}^k E_i$.

1. $\mathcal{M}_i$ admits $O(1)$-competitive algorithm (Easy parts).
2. Union of independent sets in each $\mathcal{M}_i$ is independent in $\mathcal{M}$. $\mathcal{I}(\bigoplus_{i=1}^k \mathcal{M}_i) \subseteq \mathcal{I}(\mathcal{M})$. (Combine nicely).
3. Optimum in $\bigoplus_{i=1}^k \mathcal{M}_i$ is comparable with Optimum in $\mathcal{M}$. (Don't lose much).



(Easiest matroids): Uniform. [Independent sets = Sets of size $\leq r$.]

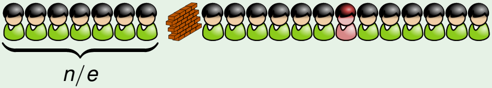
For $r = 1$: Dynkin's Algorithm



- Observe n/e objects. Accept the first record after that.

Top weight is selected w.p. $\geq 1/e$.

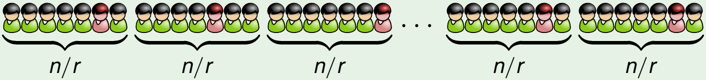
For $r = 1$: Dynkin's Algorithm



- Observe n/e objects. Accept the first record after that.

Top weight is selected w.p. $\geq 1/e$.

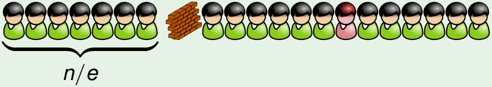
General r



- Divide in r classes and apply Dynkin's algorithm in each class.

(Easiest matroids): Uniform. [Independent sets = Sets of size $\leq r$.]

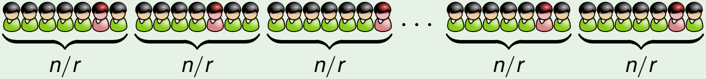
For $r = 1$: Dynkin's Algorithm



- Observe n/e objects. Accept the first record after that.

Top weight is selected w.p. $\geq 1/e$.

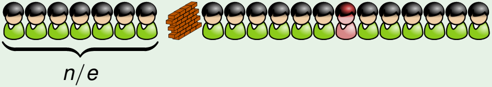
General r



- Divide in r classes and apply Dynkin's algorithm in each class.
- Each of the r top weights is the best of its class with prob. $\geq (1 - 1/r)^{r-1} \geq C > 0$. Thus it is selected with prob. $\geq C/e$.

(Easiest matroids): Uniform. [Independent sets = Sets of size $\leq r$.]

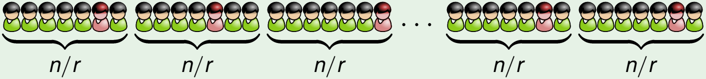
For $r = 1$: Dynkin's Algorithm



- Observe n/e objects. Accept the first record after that.

Top weight is selected w.p. $\geq 1/e$.

General r



- Divide in r classes and apply Dynkin's algorithm in each class.
- Each of the r top weights is the best of its class with prob. $\geq (1 - 1/r)^{r-1} \geq C > 0$. Thus it is selected with prob. $\geq C/e$.
- e/C (constant) competitive algorithm.

A loopless matroid is uniformly dense if

$$\frac{|F|}{\mathrm{rk}(F)} \leq \frac{|E|}{\mathrm{rk}(E)}, \text{ for all } F \neq \emptyset.$$

A loopless matroid is uniformly dense if

$$\frac{|F|}{\mathrm{rk}(F)} \leq \frac{|E|}{\mathrm{rk}(E)}, \text{ for all } F \neq \emptyset.$$

Property: Sets of $\mathrm{rk}(E)$ elements have almost full rank.

$$\mathbb{E}_{(X:|X|=\mathrm{rk}(E))}[\mathrm{rk}(X)] \geq \mathrm{rk}(E)(1 - 1/e).$$

(Easy matroids): Uniformly dense matroids are like Uniform

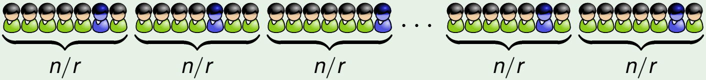
A loopless matroid is uniformly dense if

$$\frac{|F|}{\mathrm{rk}(F)} \leq \frac{|E|}{\mathrm{rk}(E)}, \text{ for all } F \neq \emptyset.$$

Property: Sets of $\mathrm{rk}(E)$ elements have almost full rank.

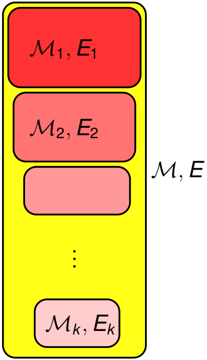
$$\mathbb{E}_{(X:|X|=\mathrm{rk}(E))}[\mathrm{rk}(X)] \geq \mathrm{rk}(E)(1-1/e).$$

Algorithm: Simulate e/C-comp. alg. for Uniform Matroids.



- Try to add each selected weight to the independent set.
- Selected elements have expected rank $\geq r(1-1/e)$.
- We recover $(1-1/e) \cdot C/e$ fraction of the top r weights.

Uniformly Dense (sub)matroids That combine nicely



Want:

Matroids $\mathcal{M}_1, \ldots, \mathcal{M}_k$ such that:

1. Each $\mathcal{M}_i$ is uniformly dense.
2. If $I_i \in \mathcal{I}(\mathcal{M}_i)$, then $I_1 \cup I_2 \cup \cdots \cup I_k \in \mathcal{I}(\mathcal{M})$. (combine nicely)

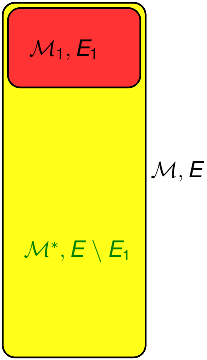
Uniformly Dense (sub)matroids That combine nicely

Procedure.

- Let E_1 be the densest set of $\mathcal{M}$ of maximum cardinality.

$$\gamma(\mathcal{M}) := \max_{F \subseteq E} \frac{|F|}{\mathrm{rk}_{\mathcal{M}}(F)} = \frac{|E_1|}{\mathrm{rk}_{\mathcal{M}}(E_1)}.$$

- $\mathcal{M}_1 = \mathcal{M}|_{E_1}$ is uniformly dense.
- $\mathcal{M}^* = \mathcal{M}/E_1$ loopless and combine nicely with $\mathcal{M}_1$.



Want:

Matroids $\mathcal{M}_1, \ldots, \mathcal{M}_k$ such that:

1. Each $\mathcal{M}_i$ is uniformly dense.
2. If $I_i \in \mathcal{I}(\mathcal{M}_i)$, then $I_1 \cup I_2 \cup \cdots \cup I_k \in \mathcal{I}(\mathcal{M})$. (combine nicely)

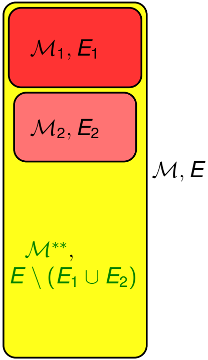
Uniformly Dense (sub)matroids That combine nicely

Procedure.

- Let E_1 be the densest set of $\mathcal{M}$ of maximum cardinality.

$$\gamma(\mathcal{M}) := \max_{F \subseteq E} \frac{|F|}{\mathrm{rk}_{\mathcal{M}}(F)} = \frac{|E_1|}{\mathrm{rk}_{\mathcal{M}}(E_1)}.$$

- $\mathcal{M}_1 = \mathcal{M}|_{E_1}$ is uniformly dense.
- $\mathcal{M}^* = \mathcal{M}/E_1$ loopless and combine nicely with $\mathcal{M}_1$.
- Iterate on $\mathcal{M}^*$....



Want:

Matroids $\mathcal{M}_1, \ldots, \mathcal{M}_k$ such that:

1. Each $\mathcal{M}_i$ is uniformly dense.
2. If $I_i \in \mathcal{I}(\mathcal{M}_i)$, then $I_1 \cup I_2 \cup \cdots \cup I_k \in \mathcal{I}(\mathcal{M})$. (combine nicely)

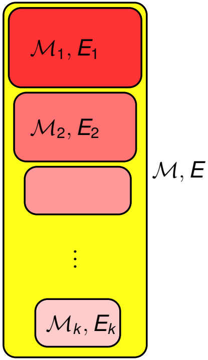
Uniformly Dense (sub)matroids That combine nicely

Procedure.

- Let E_1 be the densest set of $\mathcal{M}$ of maximum cardinality.

$$\gamma(\mathcal{M}) := \max_{F \subseteq E} \frac{|F|}{\mathrm{rk}_{\mathcal{M}}(F)} = \frac{|E_1|}{\mathrm{rk}_{\mathcal{M}}(E_1)}.$$

- $\mathcal{M}_1 = \mathcal{M}|_{E_1}$ is uniformly dense.
- $\mathcal{M}^* = \mathcal{M}/E_1$ loopless and combine nicely with $\mathcal{M}_1$.
- Iterate on $\mathcal{M}^*$....



Theorem (Principal Partition / Minors) [Tomizawa, Narayanan]

There exists a partition $E = \bigcup_{i=1}^{k} E_i$ such that

1. Each principal minor $\mathcal{M}_i = (\mathcal{M}/E_{i-1})|_{E_i}$ is uniformly dense.
2. If $I_i \in \mathcal{I}(\mathcal{M}_i)$, then $I_1 \cup I_2 \cup \cdots \cup I_k \in \mathcal{I}(\mathcal{M})$.

Algorithm for a General Matroid $\mathcal{M}$

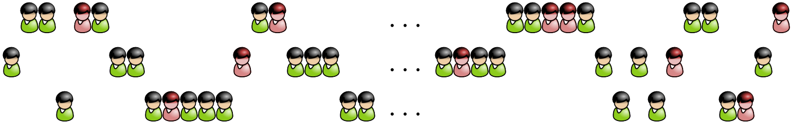
Algorithm

1. Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ be the principal minors.
2. In each $\mathcal{M}_i$ use the $O(1)$-competitive algorithm for uniformly dense matroids to obtain an independent set I_i.
3. Return $\mathrm{ALG} = I_1 \cup I_2 \cup \cdots \cup I_k$.

Algorithm for a General Matroid $\mathcal{M}$

Algorithm

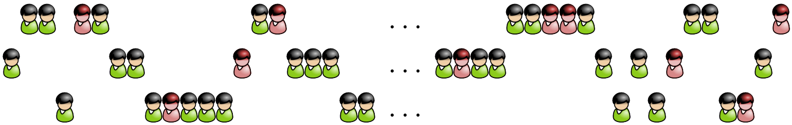
1. Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ be the principal minors.
2. In each $\mathcal{M}_i$ use the $O(1)$-competitive algorithm for uniformly dense matroids to obtain an independent set I_i.
3. Return $\text{ALG} = I_1 \cup I_2 \cup \cdots \cup I_k$.



Algorithm for a General Matroid $\mathcal{M}$

Algorithm

1. Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ be the principal minors.
2. In each $\mathcal{M}_i$ use the $O(1)$-competitive algorithm for uniformly dense matroids to obtain an independent set I_i.
3. Return $\mathrm{ALG} = I_1 \cup I_2 \cup \cdots \cup I_k$.



We have:

$$\mathbb{E}_{\pi,\sigma}[w(\mathrm{ALG})] \geq \Omega(1)\mathbb{E}_{\sigma}[w(\mathrm{OPT}_{\bigoplus \mathcal{M}_i})].$$

Also show $\mathbb{E}_{\sigma}[w(\mathrm{OPT}_{\bigoplus \mathcal{M}_i})] \geq 1/(1-1/e)\mathbb{E}_{\sigma}[w(\mathrm{OPT}_{\mathcal{M}})]$.

Summary of results.

Summary

- First constant competitive algorithm for Matroid Secretary Problem in Random-Assign. Random-Order Model.
- [OG-V] uses the same ideas for Random-Assign. Adv.-Order Model.
- Algorithm only makes comparisons.

Results

		Order	
		ADV.	RAND.
Assign.	ADV.	n	$O(\log r)$
	RAND.	$40/(1-1/e)$ [OG-V11] $16/(1-1/e)$ [S11b]	$2e/(1-1/e)$ [S11a] 5.7187 [S11b]

PP. of a matroid $\mathcal{M} = (E, \mathcal{I})$

For $\lambda \geq 0$, let $f_\lambda(X) = \lambda r(X) + |E \setminus X|$ and $\mathcal{D}_\lambda$, the minimizers of f_λ.

PP. of a matroid $\mathcal{M} = (E, \mathcal{I})$

For $\lambda \geq 0$, let $f_\lambda(X) = \lambda r(X) + |E \setminus X|$ and $\mathcal{D}_\lambda$, the minimizers of f_λ.

1. f_λ is submodular.

PP. of a matroid $\mathcal{M} = (E, \mathcal{I})$

For $\lambda \geq 0$, let $f_\lambda(X) = \lambda r(X) + |E \setminus X|$ and $\mathcal{D}_\lambda$, the minimizers of f_λ.

1. f_λ is submodular.
2. $\mathcal{D}_\lambda$ is a distributive lattice: Unique minimal F_λ^- and maximal F_λ^+.

Principal partition of matroids and submodular systems.

[Bruno, Weiberg, Tomizawa, Narayanan, Iri, Nakamura, Fujishige,...]

PP. of a matroid $\mathcal{M} = (E, \mathcal{I})$

For $\lambda \geq 0$, let $f_\lambda(X) = \lambda r(X) + |E \setminus X|$ and $\mathcal{D}_\lambda$, the minimizers of f_λ.

1. f_λ is submodular.
2. $\mathcal{D}_\lambda$ is a distributive lattice: Unique minimal F_λ^- and maximal F_λ^+.
3. P.P.: Union of all $\mathcal{D}_\lambda$.

Principal partition of matroids and submodular systems.

[Bruno, Weiberg, Tomizawa, Narayanan, Iri, Nakamura, Fujishige,...]

PP. of a matroid $\mathcal{M} = (E, \mathcal{I})$

For $\lambda \geq 0$, let $f_\lambda(X) = \lambda r(X) + |E \setminus X|$ and $\mathcal{D}_\lambda$, the minimizers of f_λ.

1. f_λ is submodular.
2. $\mathcal{D}_\lambda$ is a distributive lattice: Unique minimal F_λ^- and maximal F_λ^+.
3. P.P.: Union of all $\mathcal{D}_\lambda$.
4. Let $\lambda_1 > \cdots > \lambda_k \geq 0$ be the *critical values* s.t. $F_\lambda^- \neq F_\lambda^+$, then:

$$\emptyset = F_{\lambda_1}^- \subset (F_{\lambda_1}^+ = F_{\lambda_2}^-) \subset \cdots \subset (F_{\lambda_{k-1}}^+ = F_{\lambda_k}^-) \subset F_{\lambda_k}^+ = E.$$

PP. of a matroid $\mathcal{M} = (E, \mathcal{I})$

For $\lambda \geq 0$, let $f_\lambda(X) = \lambda r(X) + |E \setminus X|$ and $\mathcal{D}_\lambda$, the minimizers of f_λ.

1. f_λ is submodular.
2. $\mathcal{D}_\lambda$ is a distributive lattice: Unique minimal F_λ^- and maximal F_λ^+.
3. P.P.: Union of all $\mathcal{D}_\lambda$.
4. Let $\lambda_1 > \cdots > \lambda_k \geq 0$ be the *critical values* s.t. $F_\lambda^- \neq F_\lambda^+$, then:

$$\emptyset = F_{\lambda_1}^- \subset (F_{\lambda_1}^+ = F_{\lambda_2}^-) \subset \cdots \subset (F_{\lambda_{k-1}}^+ = F_{\lambda_k}^-) \subset F_{\lambda_k}^+ = E.$$

5. $\mathcal{M}_i = (\mathcal{M}|_{F_{\lambda_i}})/F_{\lambda_{i-1}}$ is uniformly dense (same as before).

PP. of a matroid $\mathcal{M} = (E, \mathcal{I})$

For $\lambda \geq 0$, let $f_\lambda(X) = \lambda r(X) + |E \setminus X|$ and $\mathcal{D}_\lambda$, the minimizers of f_λ.

1. f_λ is submodular.
2. $\mathcal{D}_\lambda$ is a distributive lattice: Unique minimal F_λ^- and maximal F_λ^+.
3. P.P.: Union of all $\mathcal{D}_\lambda$.
4. Let $\lambda_1 > \cdots > \lambda_k \geq 0$ be the *critical values* s.t. $F_\lambda^- \neq F_\lambda^+$, then:

$$\emptyset = F_{\lambda_1}^- \subset (F_{\lambda_1}^+ = F_{\lambda_2}^-) \subset \cdots \subset (F_{\lambda_{k-1}}^+ = F_{\lambda_k}^-) \subset F_{\lambda_k}^+ = E.$$

5. $\mathcal{M}_i = (\mathcal{M}|_{F_{\lambda_i}})/F_{\lambda_{i-1}}$ is uniformly dense (same as before).

Extension: Similar partition is obtained for

$f_\lambda(X) = \lambda \rho_1(X) + \rho_2(E \setminus X)$, where ρ_i are polymatroid rank fns.

Open problems

Matroid Secretary Problems

- Find $O(1)$-comp. algorithm for Adv.-Assign. Random-Order Model.
- (Weaker) What if one can choose the next element being revealed?

Principal Partition (and general secretary problems)

- [BIK07]: Generalized secretary problem for hereditary domains.
  No $o(\log n / \log\log n)$-comp. algorithm.
  Can we attain $O(\log n)$ (or even $o(n)$)?.
- *Approximate* notion of PP for non-matroidal domains?
  E.g. Matroid Intersection, p-independence systems.