An Introduction to Matroids & Greedy in Approximation Algorithms

(Julià̀n Mestre, ESA 2006)

Subset systems

Subset systems

- Let E be a finite set. Let $\mathcal{L}$ be a non-empty family of subsets of E (independent sets).

Subset systems

- Let E be a finite set. Let $\mathcal{L}$ be a non-empty family of subsets of E (independent sets).
- Def. $(E, \mathcal{L})$ is called a subset system if:

$$\forall A \in \mathcal{L},\ \ \forall A' \subseteq A,\ \ A' \in \mathcal{L}$$

[hereditary property]

Subset systems

- Let E be a finite set. Let $\mathcal{L}$ be a non-empty family of subsets of E (independent sets).
- Def. $(E, \mathcal{L})$ is called a subset system if:

$$\forall A \in \mathcal{L},\ \ \forall A' \subseteq A,\ \ A' \in \mathcal{L}$$

[hereditary property]

- A (positive) weight function w defined on E induces a weight function defined on $\mathcal{L}$:

$$w(X) = \sum_{e \in X} w(e)$$

Picking a heaviest independent set

- Prob. Given a subset system $(E, \mathcal{L})$ and a weight function $w : E \longrightarrow \mathbb{R}^+$, pick a maximum-weight element of $\mathcal{L}$.

Picking a heaviest independent set

- Prob. Given a subset system $(E, \mathcal{L})$ and a weight function $w : E \longrightarrow \mathbb{R}^+$, pick a maximum-weight element of $\mathcal{L}$.
- Alg. Greedy:

  $\text{SOL} \leftarrow \emptyset$

  for each $e \in E$ in non-increasing order of $w(e)$

      if $\text{SOL} + e \in \mathcal{L}$ then $\text{SOL} \leftarrow \text{SOL} + e$

  return SOL

Picking a heaviest independent set

- Prob. Given a subset system $(E, \mathcal{L})$ and a weight function $w : E \longrightarrow \mathbb{R}^+$, pick a maximum-weight element of $\mathcal{L}$.
- Alg. Greedy:

  SOL $\leftarrow \emptyset$

  for each $e \in E$ in non-increasing order of $w(e)$

      if SOL $+ e \in \mathcal{L}$ then SOL $\leftarrow$ SOL $+ e$

  return SOL
- Thm. Greedy is optimal for any weight function on $(E, \mathcal{L})$ iff $(E, \mathcal{L})$ is a matroid. (Rado-Edmonds)

What on earth is a matroid?

- Def. A subset system $(E, \mathcal{L})$ is a matroid if:

$$\forall A \in \mathcal{L},\ \ \forall B \in \mathcal{L} \text{ with } |A| < |B|,$$
$$\exists z \in B \setminus A \text{ such that } A + z \in \mathcal{L}$$

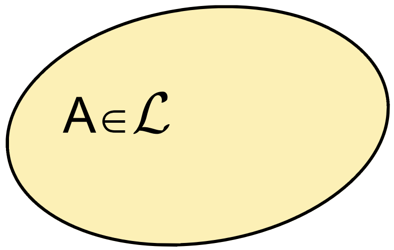
[augmentation property]

What on earth is a matroid?

- Def. A subset system $(E, \mathcal{L})$ is a matroid if:

$$\forall A \in \mathcal{L},\ \forall B \in \mathcal{L} \text{ with } |A| < |B|,$$
$$\exists z \in B \setminus A \text{ such that } A + z \in \mathcal{L}$$

[augmentation property]

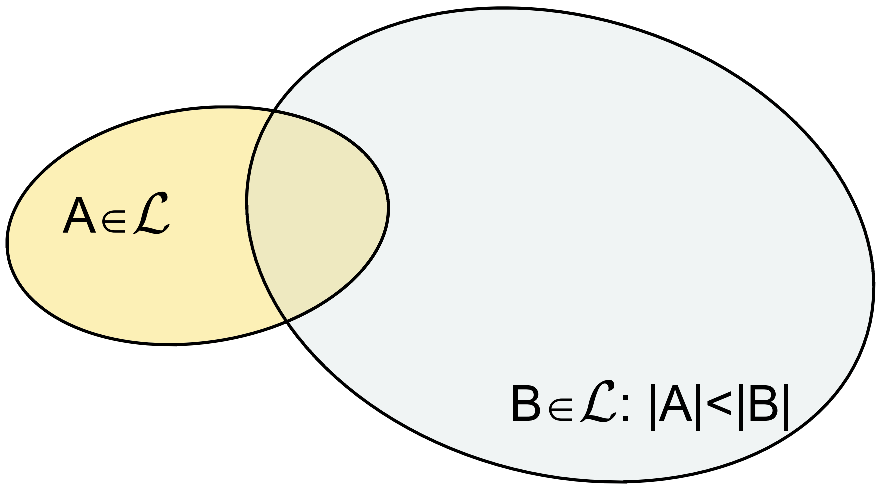


What on earth is a matroid?

- Def. A subset system $(E, \mathcal{L})$ is a matroid if:

$$\forall A \in \mathcal{L},\ \ \forall B \in \mathcal{L} \text{ with } |A| < |B|,$$
$$\exists z \in B \setminus A \text{ such that } A + z \in \mathcal{L}$$

[augmentation property]

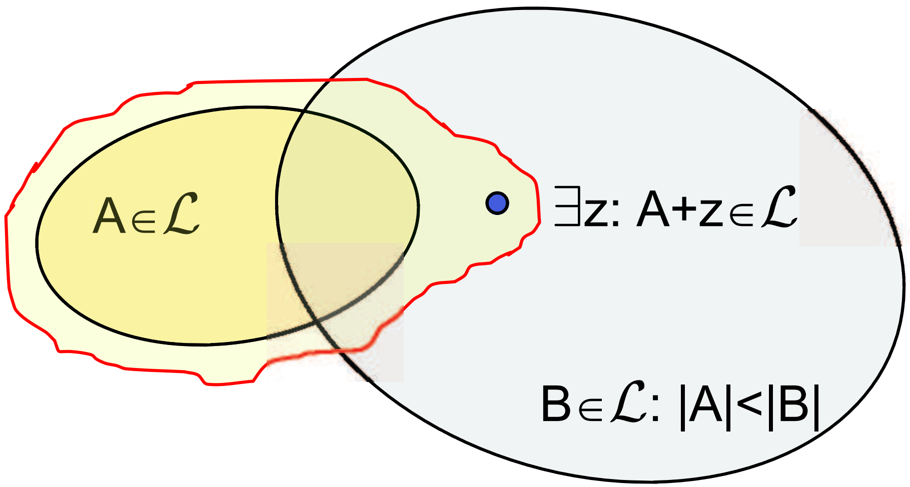


What on earth is a matroid?

- Def. A subset system $(E, \mathcal{L})$ is a matroid if:

$$\forall A \in \mathcal{L},\ \forall B \in \mathcal{L} \text{ with } |A| < |B|,$$
$$\exists z \in B \setminus A \text{ such that } A + z \in \mathcal{L}$$

[augmentation property]



Matroid examples

- Ex. 1 Subsets of at most k elements.
  - $k \in \mathbb{N}$.
  - E: finite set.
  - $\mathcal{L} = \{X \subseteq E : |X| \leq k\}$.

Matroid examples

- Ex. 1 Subsets of at most k elements.
  - $k \in \mathbb{N}$.
  - E: finite set.
  - $\mathcal{L} = \{X \subseteq E : |X| \leq k\}$.

non-empty:

$$|\emptyset| \leq k \Rightarrow \emptyset \in \mathcal{L}$$

Matroid examples

- Ex. 1 Subsets of at most k elements.
  - $k \in \mathbb{N}$.
  - E: finite set.
  - $\mathcal{L} = \{X \subseteq E : |X| \leq k\}$.

hereditary:

if $A \in \mathcal{L}$ and $A' \subseteq A$, then:

$$|A'| \leq |A| \leq k \Rightarrow A' \in \mathcal{L}$$

Matroid examples

- Ex. 1 Subsets of at most k elements.
  - $k \in \mathbb{N}$.
  - E: finite set.
  - $\mathcal{L} = \{X \subseteq E : |X| \leq k\}$.

augmentation:

if $|A| < |B|$, then for arbitrary $z \in B \setminus A$:

$$|A + z| = |A| + 1 \leq |B| - 1 + 1 \leq k$$

thus, $|A + z| \in \mathcal{L}$

Matroid examples

- Ex. 1 Subsets of at most k elements.
  - $k \in \mathbb{N}$.
  - E: finite set.
  - $\mathcal{L} = \{X \subseteq E : |X| \leq k\}$.

- Cor. We can find a heaviest subset of k elements using the Greedy algorithm.

Matroid examples (cont'd)

- Ex. 2 Column matroids.
  - $\mathbf{A}$: matrix with elements from a field.
  - $E = \{\vec{x} : \vec{x} \text{ is a column of } \mathbf{A}\}$.
  - $\mathcal{L} = \{X \subseteq E : X \text{ is linearly independent}\}$.

Matroid examples (cont'd)

- Ex. 2 Column matroids.
  - $\mathbf{A}$: matrix with elements from a field.
  - $E = \{\vec{x} : \vec{x}$ is a column of $\mathbf{A}\}$.
  - $\mathcal{L} = \{X \subseteq E : X$ is linearly independent$\}$.

non-empty:

the empty set is vacuously linearly independent

Matroid examples (cont'd)

- Ex. 2 Column matroids.
  - $\mathbf{A}$: matrix with elements from a field.
  - $E = \{\vec{x} : \vec{x} \text{ is a column of } \mathbf{A}\}$.
  - $\mathcal{L} = \{X \subseteq E : X \text{ is linearly independent}\}$.

hereditary:

linear dependency cannot be introduced by removing vectors

Matroid examples (cont'd)

- Ex. 2 Column matroids.
  - $\mathbf{A}$: matrix with elements from a field.
  - $E = \{\vec{x} : \vec{x}$ is a column of $\mathbf{A}\}$.
  - $\mathcal{L} = \{X \subseteq E : X$ is linearly independent$\}$.

augmentation:

if $\forall z \in B \setminus A$, $A + z \notin \mathcal{L}$, then each vector of B is linearly dependent on the vectors of A which implies $|B| \leq |A|$

Matroid examples (cont'd)

- Ex. 2 Column matroids.
  - $\mathbf{A}$: matrix with elements from a field.
  - $E = \{\vec{x} : \vec{x} \text{ is a column of } \mathbf{A}\}$.
  - $\mathcal{L} = \{X \subseteq E : X \text{ is linearly independent}\}$.

- Cor. We can find a heaviest base among the vectors of $\mathbf{A}$ using the Greedy algorithm.

Matroid examples (cont'd)

- Ex. 3 Cycle/Graphic matroids.
  - $G = (V, E)$: undirected graph.
  - E = edge set of G.
  - $\mathcal{L} = \{X \subseteq E : G_X = (V, X) \text{ is a forest}\}$.

Matroid examples (cont'd)

- Ex. 3 Cycle/Graphic matroids.
  - $G = (V, E)$: undirected graph.
  - E = edge set of G.
  - $\mathcal{L} = \{X \subseteq E : G_X = (V, X) \text{ is a forest}\}$.

non-empty:

$G_\emptyset = (V, \emptyset)$ is a forest

Matroid examples (cont'd)

- Ex. 3 Cycle/Graphic matroids.
  - $G = (V, E)$: undirected graph.
  - E = edge set of G.
  - $\mathcal{L} = \{X \subseteq E : G_X = (V, X) \text{ is a forest}\}$.

hereditary:

any subset of a forest is a forest

Matroid examples (cont'd)

- Ex. 3 Cycle/Graphic matroids.
  - $G = (V, E)$: undirected graph.
  - E = edge set of G.
  - $\mathcal{L} = \{X \subseteq E : G_X = (V, X) \text{ is a forest}\}$.

augmentation:

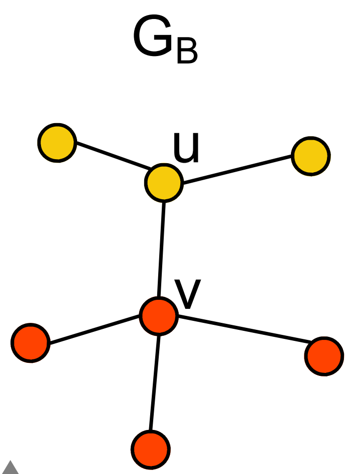
if $|A| < |B|$, then

$\#\text{trees in } G_A > \#\text{trees in } G_B$

Matroid examples (cont'd)

- Ex. 3 Cycle/Graphic matroids.
  - $G = (V, E)$: undirected graph.
  - E = edge set of G.
  - $\mathcal{L} = \{X \subseteq E : G_X = (V, X) \text{ is a forest}\}$.

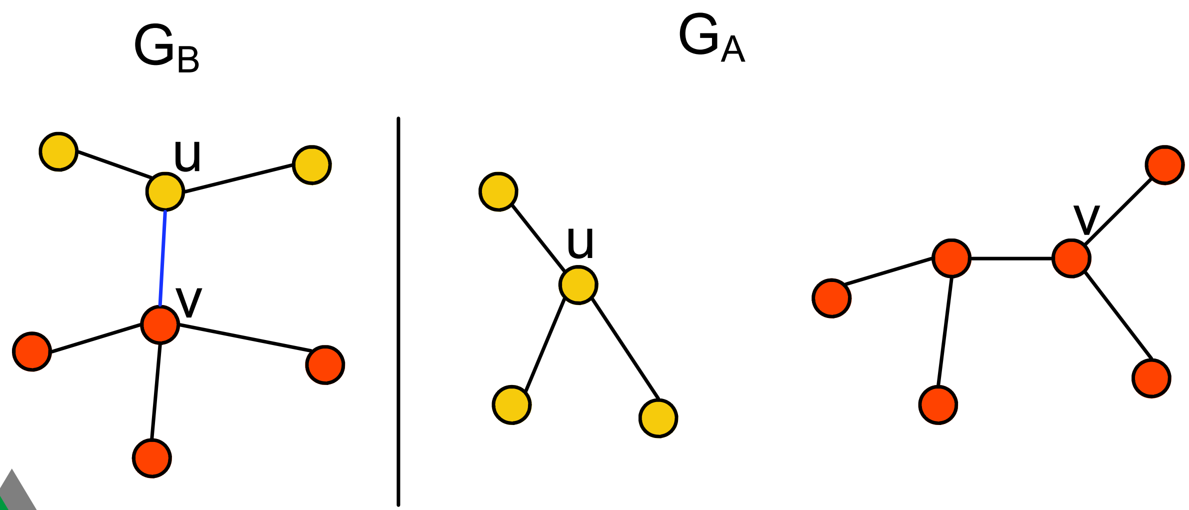
augmentation:



Matroid examples (cont'd)

- Ex. 3 Cycle/Graphic matroids.
  - $G = (V, E)$: undirected graph.
  - E = edge set of G.
  - $\mathcal{L} = \{X \subseteq E : G_X = (V, X) \text{ is a forest}\}$.

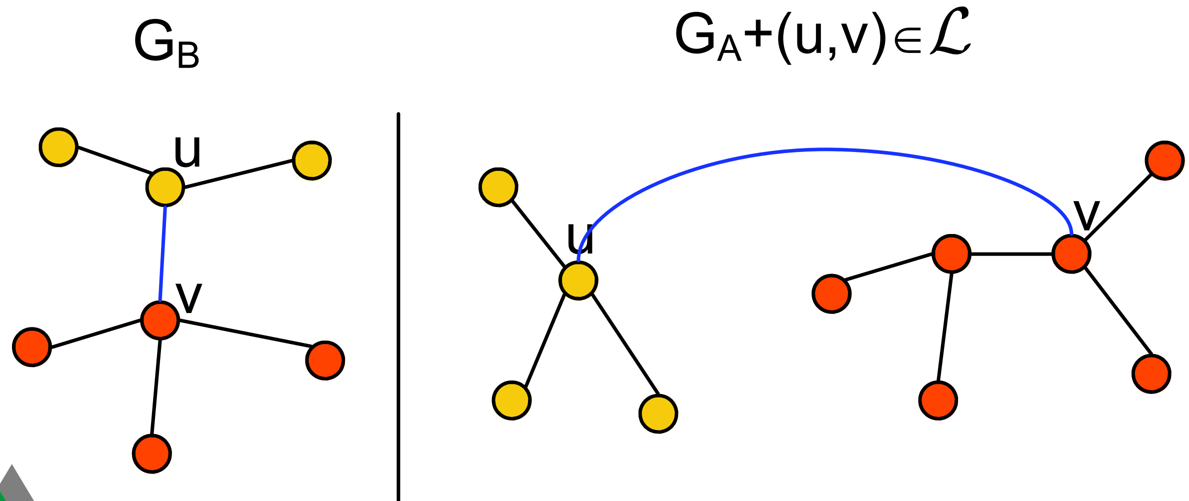
augmentation:



Matroid examples (cont'd)

- Ex. 3 Cycle/Graphic matroids.
  - $G = (V, E)$: undirected graph.
  - $E =$ edge set of G.
  - $\mathcal{L} = \{X \subseteq E : G_X = (V, X) \text{ is a forest}\}$.

augmentation:



Matroid examples (cont'd)

- Ex. 3 Cycle/Graphic matroids.
  - $G = (V, E)$: undirected graph.
  - E = edge set of G.
  - $\mathcal{L} = \{X \subseteq E : G_X = (V, X) \text{ is a forest}\}$.

- Cor. We can find a heaviest spanning tree of G using the Greedy algorithm.

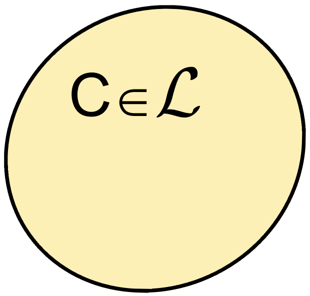
k-extendible systems

- Def. A subset system $(E, \mathcal{L})$ is k-extendible if:

$$\forall C \in \mathcal{L},\ \forall x \notin C \text{ with } C + x \in \mathcal{L},$$
$$\forall D \text{ extension of } C,\ \exists Y \subseteq D \setminus C \text{ such that}$$
$$|Y| \leq k \text{ and } D \setminus Y + x \in \mathcal{L}$$

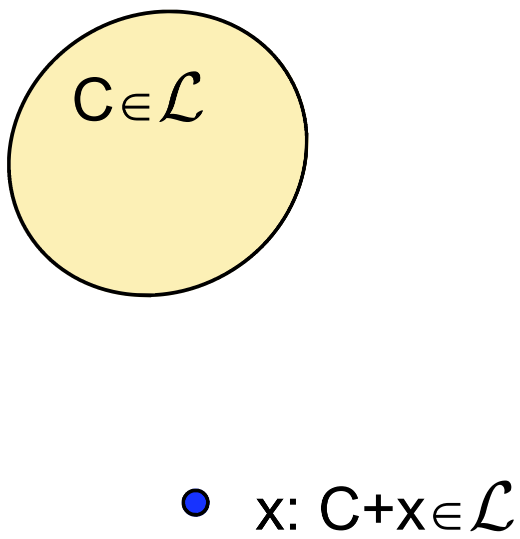
k-extendible systems

- Def. A subset system $(E, \mathcal{L})$ is k-extendible if:



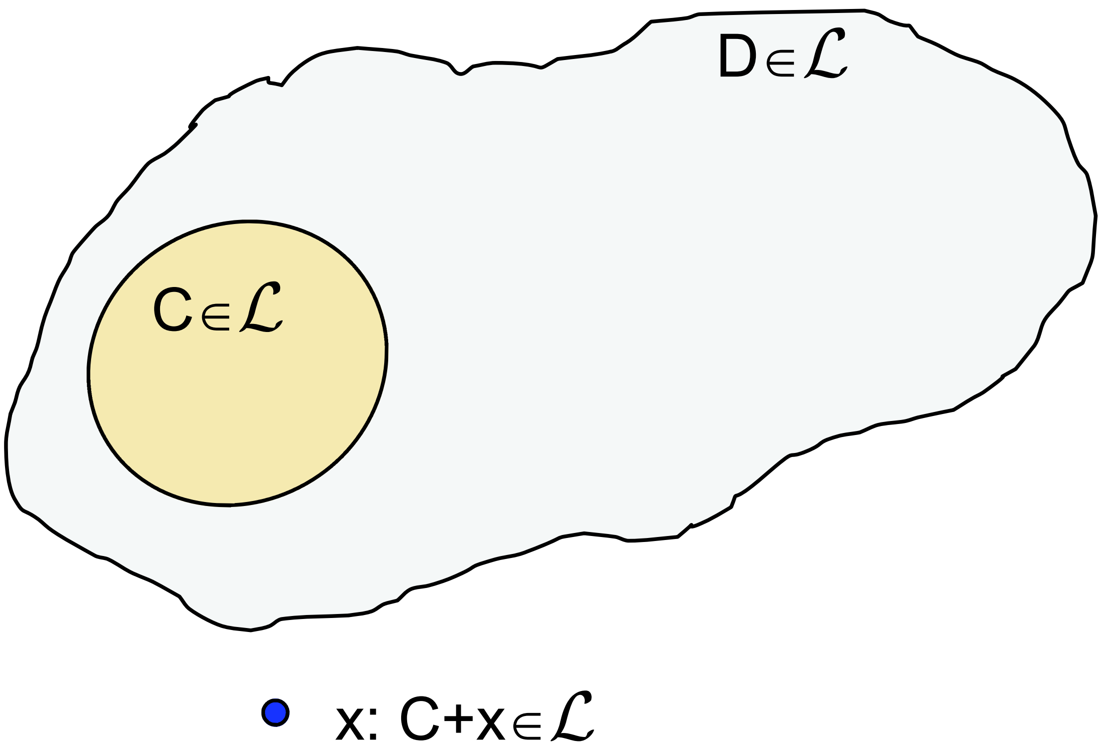
k-extendible systems

- Def. A subset system $(E, \mathcal{L})$ is k-extendible if:



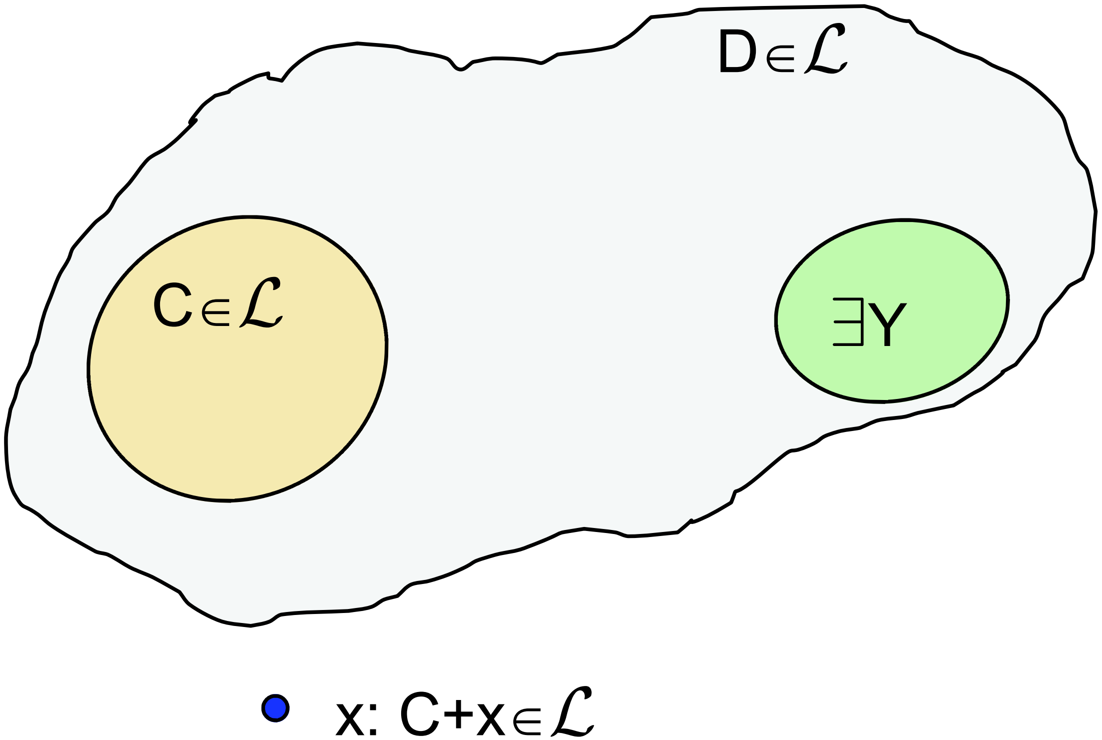
k-extendible systems

- Def. A subset system $(E, \mathcal{L})$ is k-extendible if:



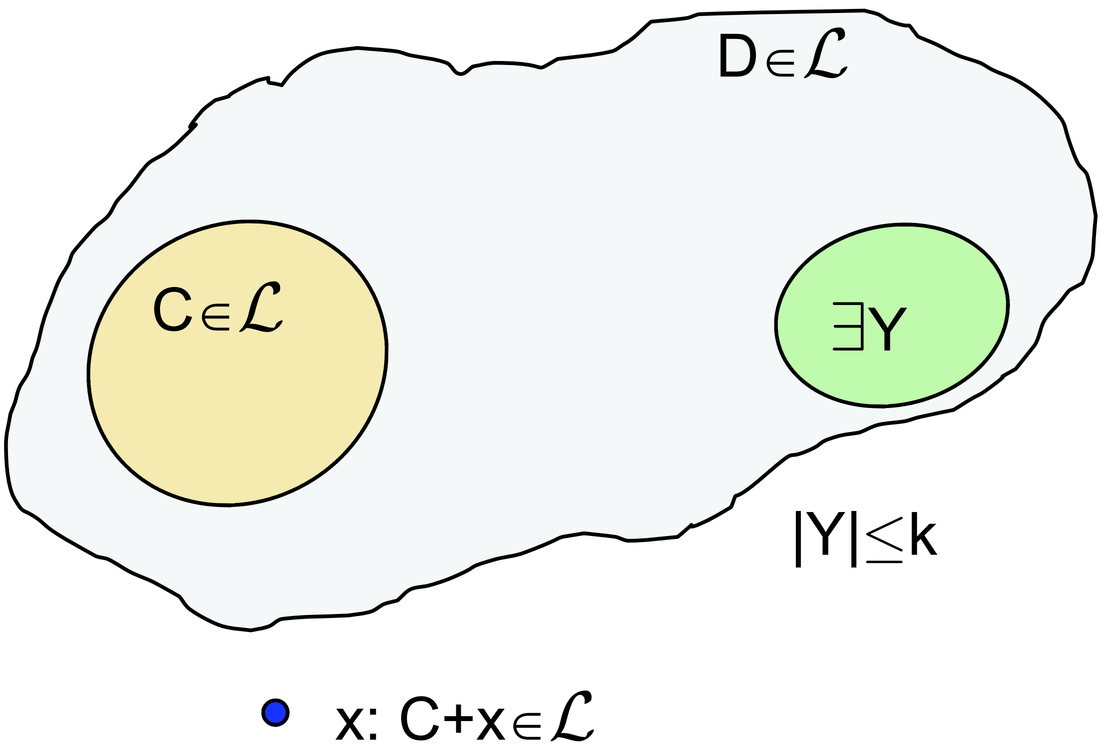
k-extendible systems

- Def. A subset system $(E, \mathcal{L})$ is k-extendible if:



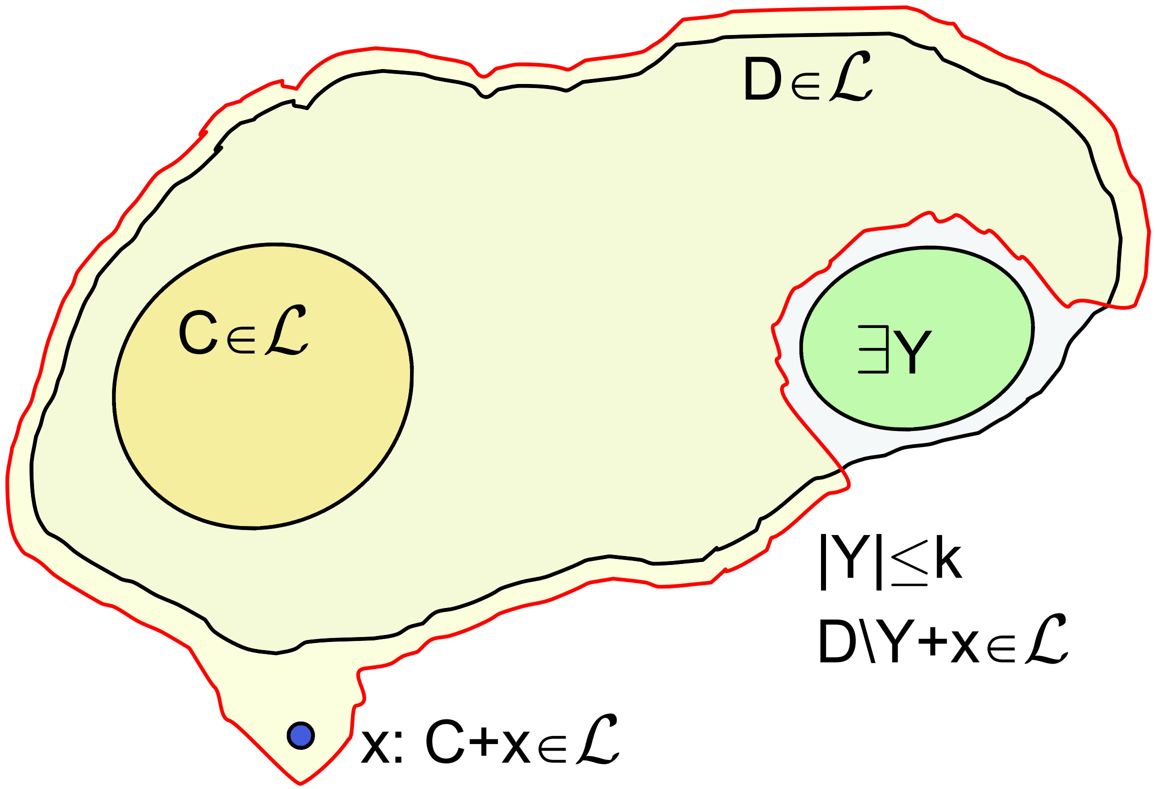
k-extendible systems

- Def. A subset system $(E, \mathcal{L})$ is k-extendible if:



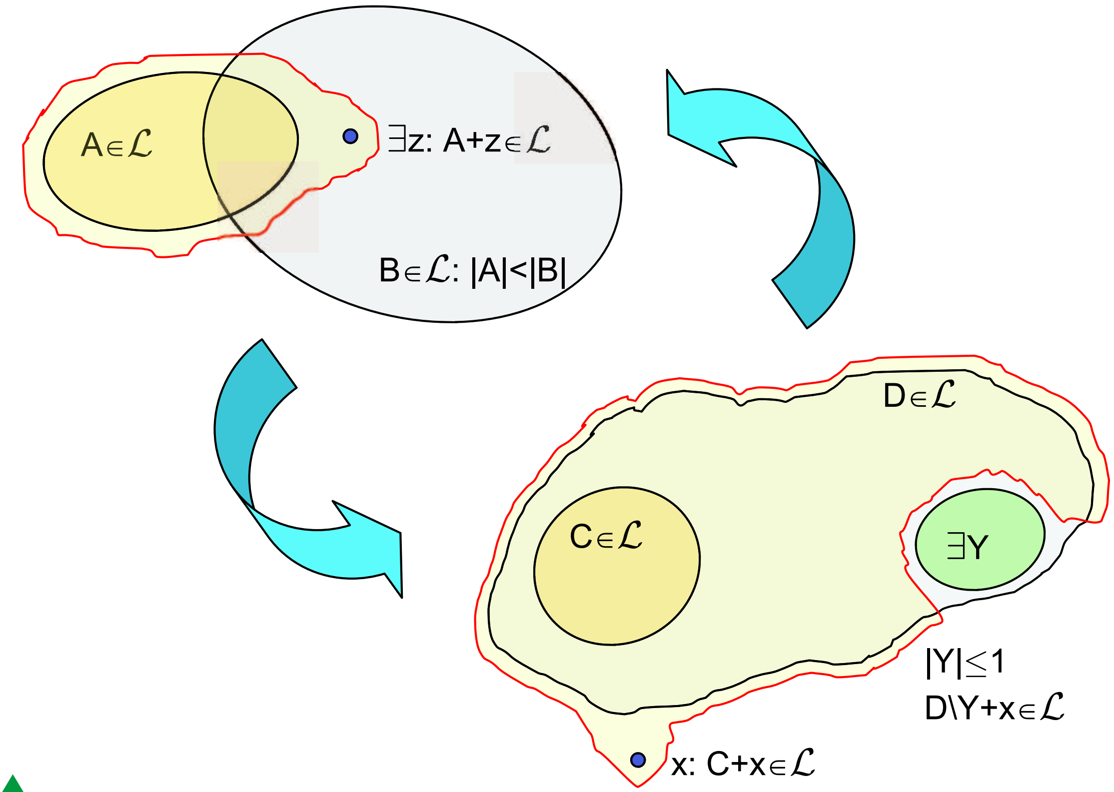
k-extendible systems

- Def. A subset system $(E, \mathcal{L})$ is k-extendible if:

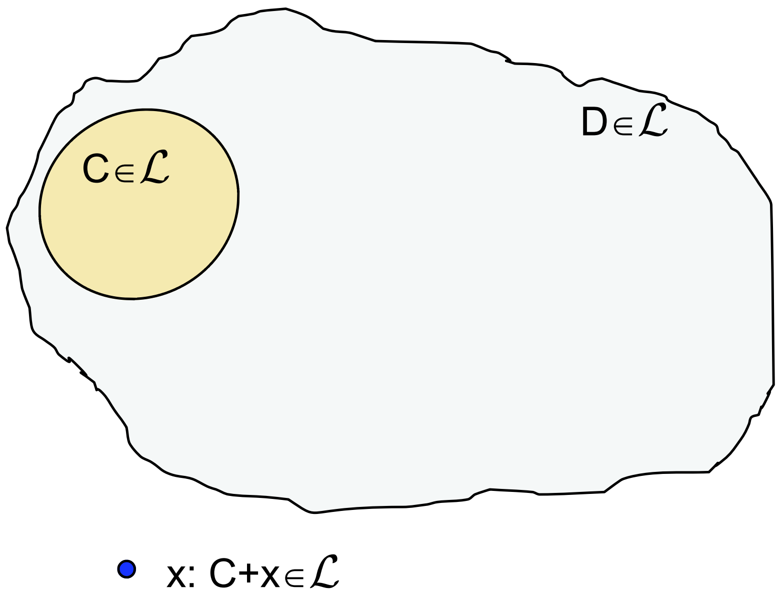


Matroids $\equiv$ 1-extendible systems

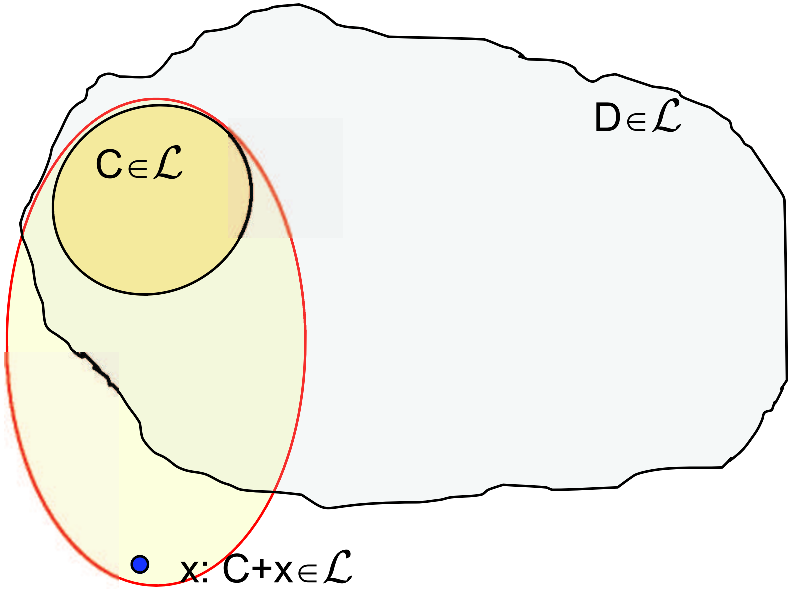
- Thm. $(E, \mathcal{L})$ is a matroid iff $(E, \mathcal{L})$ is 1-extendible.



Matroids ⊆ 1-extendible systems

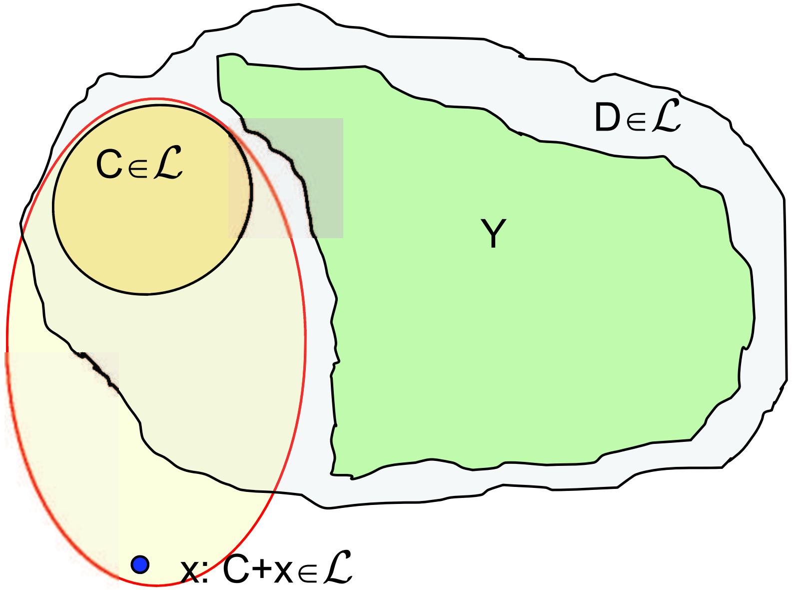


Matroids $\subseteq$ 1-extendible systems



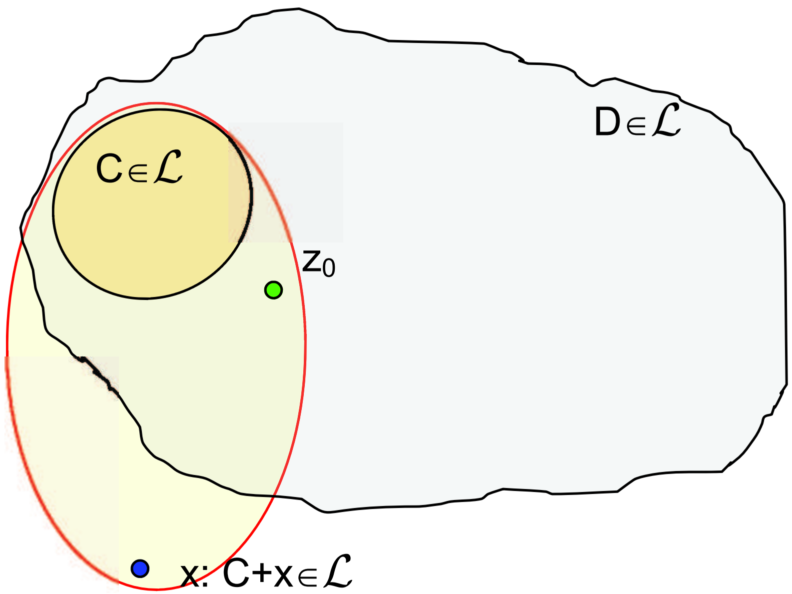
- If $|C + x| = |D|$, then $|D \setminus C| = 1$.

Matroids ⊆ 1-extendible systems



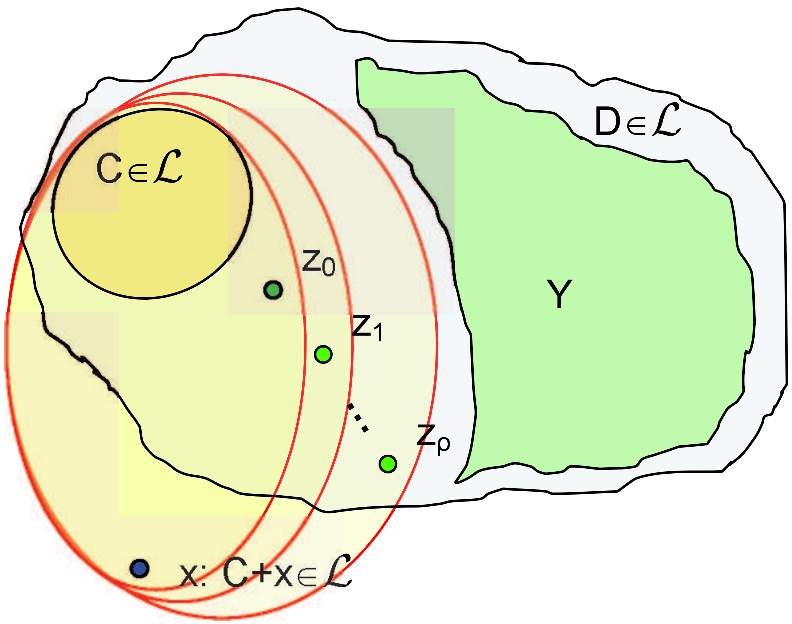
- $D \setminus Y + x \in \mathcal{L}$.

Matroids $\subseteq$ 1-extendible systems



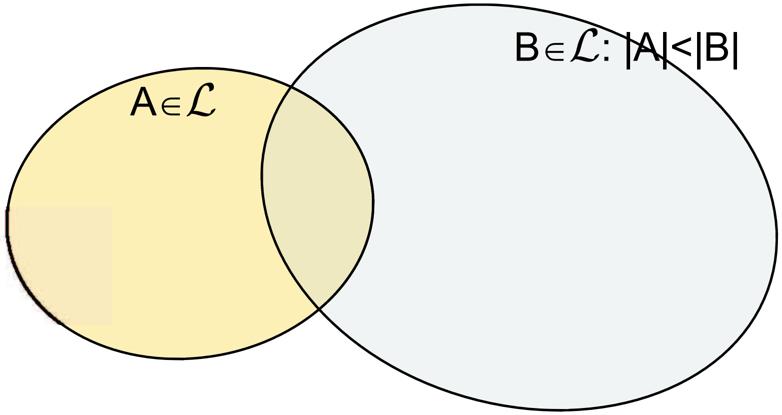
- Applying the augmentation property, $\exists z_0 \in D \setminus C$ such that $C + z_0 + x \in \mathcal{L}$.

Matroids $\subseteq$ 1-extendible systems

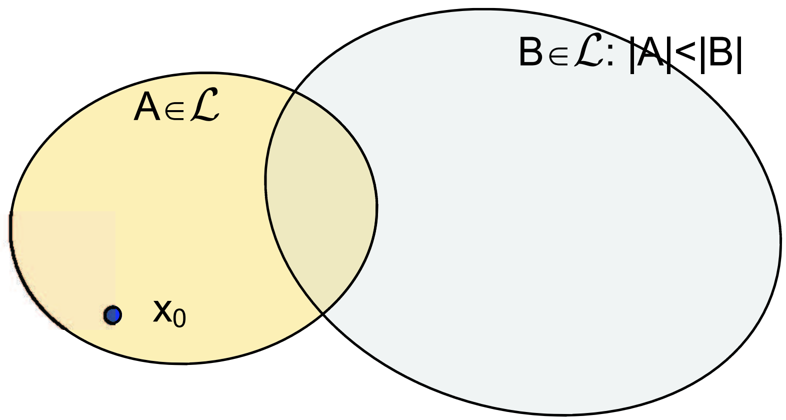


- Finally, we get a sequence $z_0, \ldots, z_\rho$ such that $C + \sum z_i + x \in \mathcal{L}$ and $|D \setminus (C + \sum z_i)| = 1$.

Matroids ⊇ 1-extendible systems

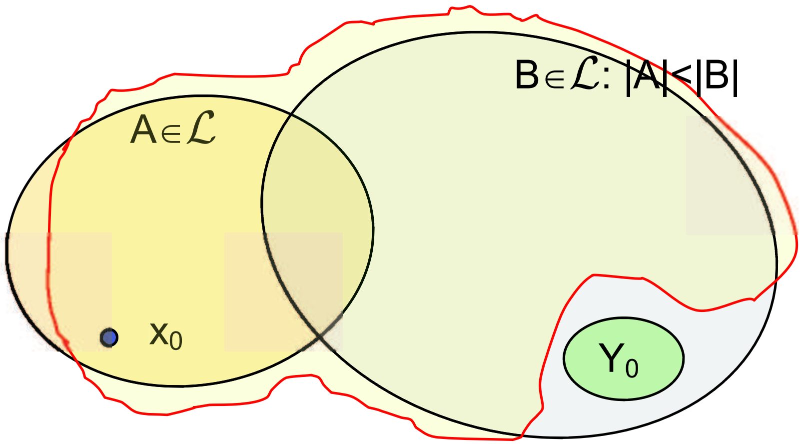


Matroids $\supseteq$ 1-extendible systems



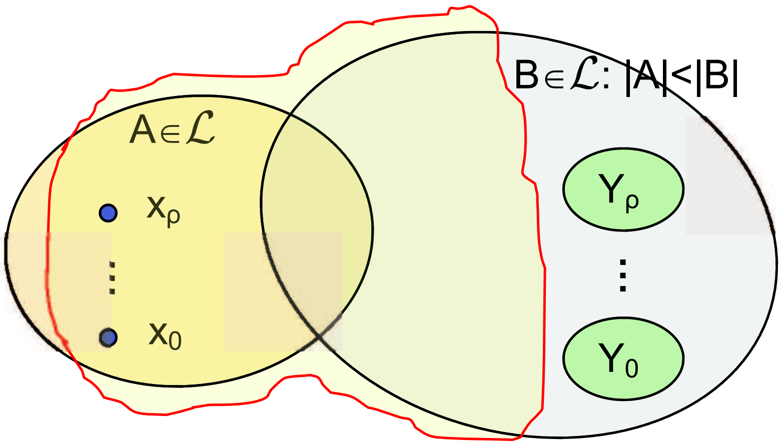
- Pick any $x_0 \in A \setminus B$.

Matroids $\supseteq$ 1-extendible systems



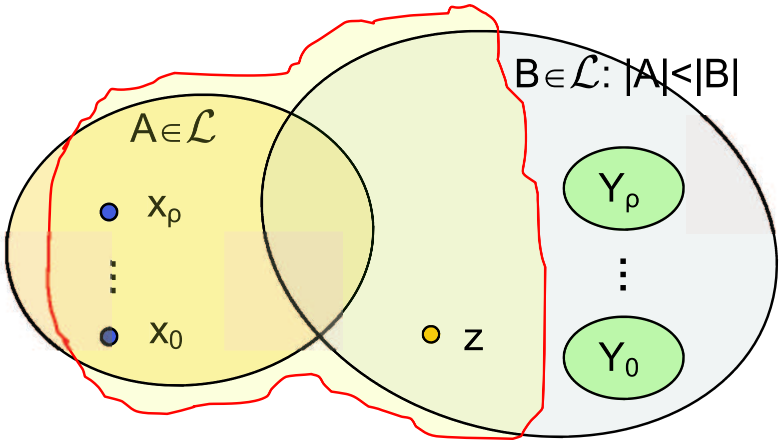
- By 1-extendibility, $\exists Y_0 \subseteq B \setminus A$ with $|Y_0| \leq 1$ such that $B \setminus Y_0 + x_0 \in \mathcal{L}$.

Matroids ⊇ 1-extendible systems



- Keep picking x_i's in $A \setminus B$ until there are no more. Then $B \setminus \bigcup Y_i + \sum x_i \in \mathcal{L}$.

Matroids $\supseteq$ 1-extendible systems



- Moreover, $A \subseteq B \setminus \bigcup Y_i + \sum x_i$, which implies $A + z \in \mathcal{L}$ for arbitrary $z \in B \setminus \bigcup Y_i$.

Greedy on k-extendible systems

- Thm. If $(E, \mathcal{L})$ is k-extendible then Greedy obtains a $\frac{1}{k}$-approximate solution for any weight function on $(E, \mathcal{L})$.

Greedy on k-extendible systems

- Thm. If $(E, \mathcal{L})$ is k-extendible then Greedy obtains a $\frac{1}{k}$-approximate solution for any weight function on $(E, \mathcal{L})$.
- Proof.
  - $x_1, x_2, \ldots, x_\rho$: successive choices of Greedy
  - $\emptyset = S_0, S_1, S_2, \ldots, S_\rho = \mathrm{SOL}$: successive partial solutions with

$$S_i = S_{i-1} + x_i \quad , \quad \forall i : 1 \leq i \leq \rho$$

Obtaining the approximation ratio

- Lemma For each $i : 1 \leq i \leq \rho$,

  $w\left(\mathrm{OPT}\left(S_{i-1}\right)\right) \leq w\left(\mathrm{OPT}\left(S_i\right)\right) + (k-1) \cdot w\left(x_i\right)$

Obtaining the approximation ratio

- Lemma For each $i : 1 \leq i \leq \rho$,

$$w\left(\mathrm{OPT}\left(S_{i-1}\right)\right) \leq w\left(\mathrm{OPT}\left(S_i\right)\right)+(k-1)\cdot w\left(x_i\right)$$

- Apply lemma ρ times:

$$\begin{aligned} \mathrm{OPT} &= w\left(\mathrm{OPT}\left(S_0\right)\right) \\ &\leq w\left(\mathrm{OPT}\left(S_\rho\right)\right) + (k-1) \cdot w\left(S_\rho\right) \\ &= k \cdot \mathrm{SOL} \end{aligned}$$

Proof of lemma

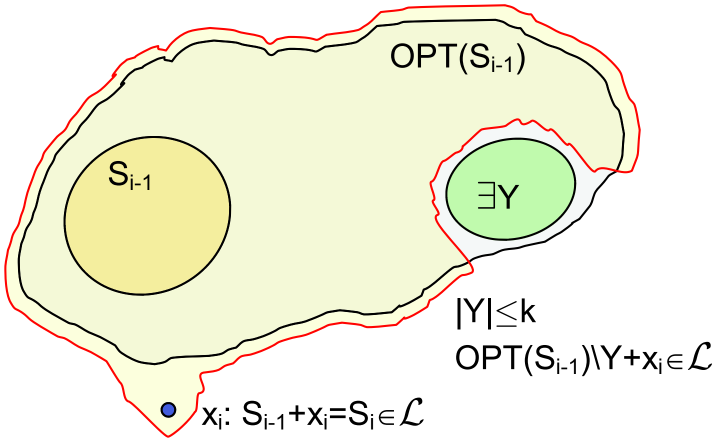
- Lemma For each $i : 1 \leq i \leq \rho$,

$$w\left(\mathrm{OPT}\left(S_{i-1}\right)\right) \leq w\left(\mathrm{OPT}\left(S_i\right)\right) + (k-1) \cdot w\left(x_i\right)$$

Proof of lemma

- Lemma For each $i : 1 \leq i \leq \rho$,

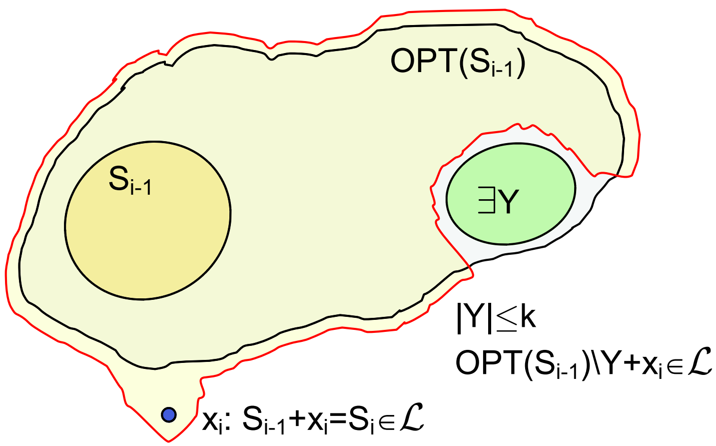
$$w\left(\mathrm{OPT}\left(S_{i-1}\right)\right) \leq w\left(\mathrm{OPT}\left(S_{i}\right)\right)+(k-1)\cdot w\left(x_{i}\right)$$



Proof of lemma

- Lemma For each $i : 1 \leq i \leq \rho$,

$$w\left(\mathrm{OPT}\left(S_{i-1}\right)\right) \leq w\left(\mathrm{OPT}\left(S_i\right)\right) + (k-1) \cdot w\left(x_i\right)$$

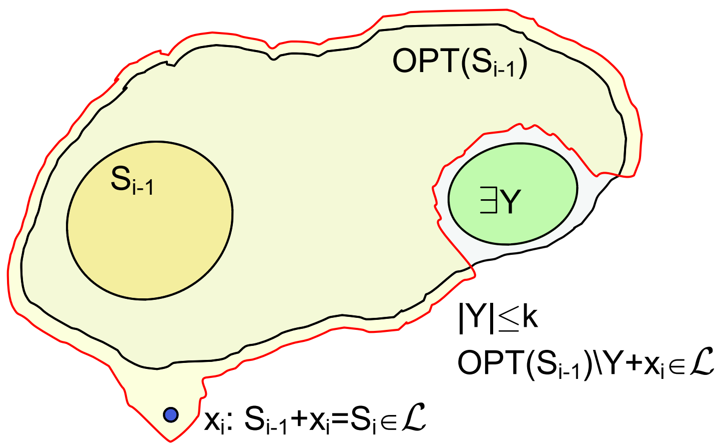


- $w\left(\mathrm{OPT}\left(S_{i-1}\right)\right) =$
  $w\left(\mathrm{OPT}\left(S_{i-1}\right) \setminus Y + x_i\right) + w\left(Y\right) - w\left(x_i\right)$

Proof of lemma

- **Lemma** For each $i : 1 \leq i \leq \rho$,

$$w\left(\mathrm{OPT}\left(S_{i-1}\right)\right) \leq w\left(\mathrm{OPT}\left(S_i\right)\right) + (k-1) \cdot w\left(x_i\right)$$

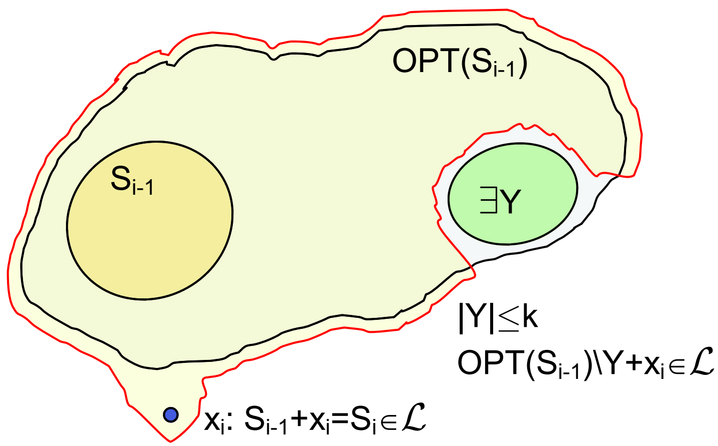


- $w\left(\mathrm{OPT}\left(S_{i-1}\right)\right) \leq w\left(\mathrm{OPT}\left(S_i\right)\right) + w\left(Y\right) - w\left(x_i\right)$

Proof of lemma

- Lemma For each $i : 1 \le i \le \rho$,

$$w\left(\mathrm{OPT}\left(S_{i-1}\right)\right) \le w\left(\mathrm{OPT}\left(S_i\right)\right) + (k-1) \cdot w\left(x_i\right)$$



- $y \in Y \Rightarrow \forall j \le i-1,\ \ S_j + y \subseteq \mathrm{OPT}\left(S_{i-1}\right) \Rightarrow$ $\forall j \le i-1,\ \ S_j + y \in \mathcal{L} \Rightarrow w\left(y\right) \le w\left(x_i\right)$

Example of a 2-extendible system

- Maximum-weight b-matching.
  - $G = (V, E)$: undirected graph, $b : V \longrightarrow \mathbb{N}$, $w : E \longrightarrow \mathbb{R}$.
  - E: edge set of G.
  - $\mathcal{L} = \{M \subseteq E : \forall u \in V,\ \deg_M(u) \leq b(u)\}$.

Example of a 2-extendible system

- Maximum-weight b-matching.
  - $G = (V, E)$: undirected graph, $b : V \longrightarrow \mathbb{N}$, $w : E \longrightarrow \mathbb{R}$.
  - E: edge set of G.
  - $\mathcal{L} = \{M \subseteq E : \forall u \in V,\ \deg_M(u) \leq b(u)\}$.

non-empty:

$$\forall u \in V,\ \deg_\emptyset(u) = 0 \leq b(u)$$

Example of a 2-extendible system

- Maximum-weight b-matching.
  - $G = (V, E)$: undirected graph, $b : V \longrightarrow \mathbb{N}$, $w : E \longrightarrow \mathbb{R}$.
  - E: edge set of G.
  - $\mathcal{L} = \{M \subseteq E : \forall u \in V,\ \deg_M(u) \leq b(u)\}$.

hereditary:

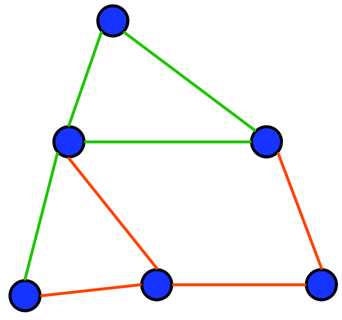
if $M \in \mathcal{L}$ and $M' \subseteq M$, then for any $u \in V$:

$$\deg_{M'}(u) \leq \deg_M(u) \leq b(u)$$

Example of a 2-extendible system

- Maximum-weight b-matching.
  - $G = (V, E)$: undirected graph, $b : V \longrightarrow \mathbb{N}$, $w : E \longrightarrow \mathbb{R}$.
  - E: edge set of G.
  - $\mathcal{L} = \{M \subseteq E : \forall u \in V,\ \deg_M(u) \leq b(u)\}$.

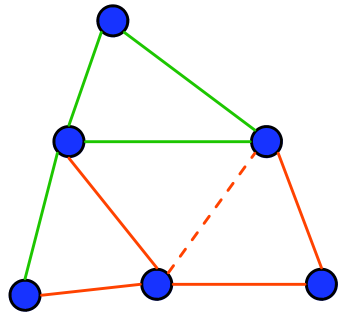
2-extendible:



Example of a 2-extendible system

- Maximum-weight b-matching.
  - $G = (V, E)$: undirected graph, $b : V \longrightarrow \mathbb{N}$, $w : E \longrightarrow \mathbb{R}$.
  - E: edge set of G.
  - $\mathcal{L} = \{M \subseteq E : \forall u \in V,\ \deg_M(u) \leq b(u)\}$.

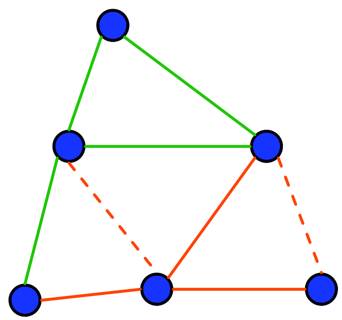
2-extendible:



Example of a 2-extendible system

- Maximum-weight b-matching.
  - $G = (V, E)$: undirected graph, $b : V \longrightarrow \mathbb{N}$, $w : E \longrightarrow \mathbb{R}$.
  - E: edge set of G.
  - $\mathcal{L} = \{M \subseteq E : \forall u \in V,\ \deg_M(u) \leq b(u)\}$.

2-extendible:



Example of a 2-extendible system

- Maximum-weight b-matching.
  - $G = (V, E)$: undirected graph, $b : V \longrightarrow \mathbb{N}$, $w : E \longrightarrow \mathbb{R}$.
  - E: edge set of G.
  - $\mathcal{L} = \{M \subseteq E : \forall u \in V,\ \deg_M(u) \leq b(u)\}$.

- Cor. Greedy is a $\frac{1}{2}$-approximation algorithm for Maximum-weight b-matching.

Other k-extendible systems

- Maximum profit scheduling (2-extendible).
- Maximum asymmetric TSP (3-extendible).
- Intersection of k matroids (k-extendible).

Tradeoffs for b-matching

- Maximum-weight b-matching can be solved exactly in time $O\left(\sum b(u) \cdot \min\left(m \log n, n^2\right)\right)$.

Tradeoffs for b-matching

- Maximum-weight b-matching can be solved exactly in time $O\left(\sum b(u) \cdot \min\left(m \log n, n^2\right)\right)$.
- Therefore the Greedy algorithm should be regarded as a tradeoff: $\frac{1}{2}$-approximation in time $O(m \log n)$.

Tradeoffs for b-matching

- Maximum-weight b-matching can be solved exactly in time $O\left(\sum b(u) \cdot \min\left(m \log n, n^2\right)\right)$.
- Therefore the Greedy algorithm should be regarded as a tradeoff: $\frac{1}{2}$-approximation in time $O(m \log n)$.
- Improvement: $\frac{1}{2}$-approximation in time $O(bm)$.

Tradeoffs for b-matching

- Maximum-weight b-matching can be solved exactly in time $O\left(\sum b(u) \cdot \min\left(m \log n, n^2\right)\right)$.
- Therefore the Greedy algorithm should be regarded as a tradeoff: $\frac{1}{2}$-approximation in time $O(m \log n)$.
- Improvement: $\frac{1}{2}$-approximation in time $O(bm)$.
- Further improvement: randomized $\left(\frac{2}{3} - \epsilon\right)$-approximation in time $O\left(bm \log \frac{1}{\epsilon}\right)$.

b-matching by greedy walks

- Alg. Find-Walk(u)

  $b(u) \leftarrow b(u) - 1$

  if $\deg(u) = 0$ **then return** $\emptyset$

  let (u, v) be the heaviest edge out of u

  remove (u, v) from G

  if $b(u) = 0$ **then** remove all edges incident to u

  return $(u, v) +$ Find-Walk(v)

b-matching by greedy walks

- Alg. Find-Walk(u)

  $b(u) \leftarrow b(u) - 1$

  if $\deg(u) = 0$ **then return** $\emptyset$

  let (u, v) be the heaviest edge out of u

  remove (u, v) from G

  if $b(u) = 0$ **then** remove all edges incident to u

  return $(u, v) +$ Find-Walk(v)

- **while** $\exists u \in V$ s.t. $b(u) > 0$ and $\deg(u) > 0$ **do**

  $M \leftarrow M +$ Find-Walk(u)

b-matching by greedy walks

- Alg. Find-Walk(u)

  $b(u) \leftarrow b(u) - 1$

  if $\deg(u) = 0$ then return $\emptyset$

  let (u, v) be the heaviest edge out of u

  remove (u, v) from G

  if $b(u) = 0$ then remove all edges incident to u

  return $(u, v) +$ Find-Walk(v)

- while $\exists u \in V$ s.t. $b(u) > 0$ and $\deg(u) > 0$ do

  $M \leftarrow M +$ Find-Walk(u)

- Only guarantees that $\deg_M(u) \leq 2b(u)$.

$\frac{1}{2}$-approximation for b-matching

- Split M into M_1 and M_2 by taking alternative edges of individual walks. Pick the heaviest of M_1, M_2.

$\frac{1}{2}$-approximation for b-matching

- Split M into M_1 and M_2 by taking alternative edges of individual walks. Pick the heaviest of M_1, M_2.
- SOL $\geq \frac{w(M)}{2} \geq \frac{w(M_{\text{OPT}})}{2}$. Why?

$\frac{1}{2}$-approximation for b-matching

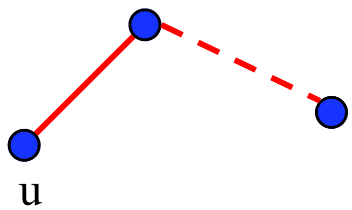
- Split M into M_1 and M_2 by taking alternative edges of individual walks. Pick the heaviest of M_1, M_2.
- SOL $\geq \frac{w(M)}{2} \geq \frac{w(M_{\mathrm{OPT}})}{2}$. Why?
- To each edge (u, v) picked by Find-Walk, assign some edge $e \in M_{\mathrm{OPT}}$.
  - If $(u, v) \in M_{\mathrm{OPT}}$, assign it to itself.
  - Otherwise, pick any $e \in M_{\mathrm{OPT}}$ incident to u.

$\frac{1}{2}$-approximation for b-matching

- Split M into M_1 and M_2 by taking alternative edges of individual walks. Pick the heaviest of M_1, M_2.
- SOL $\geq \frac{w(M)}{2} \geq \frac{w(M_{\text{OPT}})}{2}$. Why?
- To each edge (u, v) picked by Find-Walk, assign some edge $e \in M_{\text{OPT}}$.
  - If $(u, v) \in M_{\text{OPT}}$, assign it to itself.
  - Otherwise, pick any $e \in M_{\text{OPT}}$ incident to u.
- Each edge in M_{OPT} is assigned to a unique edge in M. Moreover, $w\,(e) \leq w\,(u, v)$.

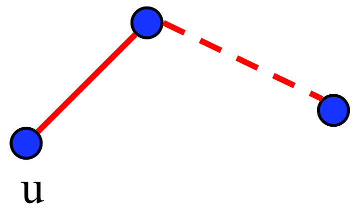
Arms and pieces

Arms and pieces

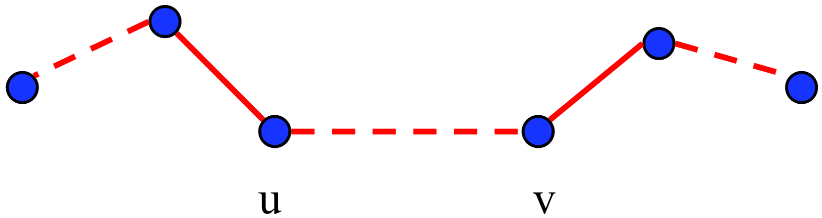


An Arm out of node u

Arms and pieces



An Arm out of node u



A Piece about edge (u, v)

A randomized algorithm for b-matching

- Alg. Linear-Random(G, w)

  $M \leftarrow \emptyset$

  repeat k times

  pick a vertex u uniformly at random

  with probability $\frac{\deg_M(u)}{b}$ do

  pick $(u, v) \in M$ uniformly at random

  find max-benefit compatible piece P about (u, v)

  $M \leftarrow M \oplus P$

  with probability $\frac{b(u) - \deg_M(u)}{b}$ do

  find max-benefit compatible arm A out of u

  $M \leftarrow M \oplus A$