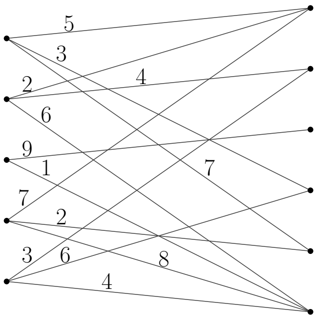
A Fast Approximation for Maximum Weight Matroid Intersection

Chandra Chekuri   *Kent Quanrud*

University of Illinois at Urbana-Champaign

January 10, 2016

Max. weight bipartite matching



Input:

bipartite graph $G = (U \sqcup V, E)$

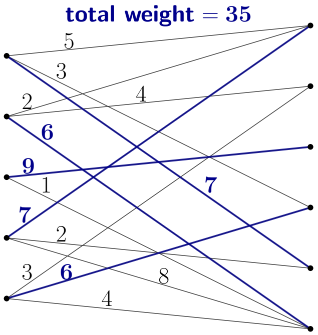
weights $w : E \to \mathbb{R}_{\geq 0}$

Output:

matching $M \subseteq E$ maximizing

$$w(M) \stackrel{\text{def}}{=} \sum_{e \in M} w(e)$$

Max. weight bipartite matching



Input:

bipartite graph $G = (U \sqcup V, E)$

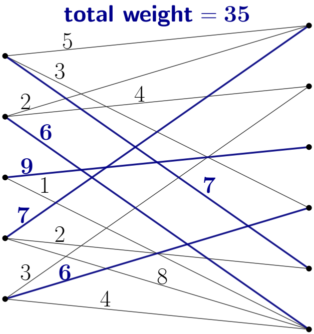
weights $w : E \to \mathbb{R}_{\geq 0}$

Output:

matching $M \subseteq E$ maximizing

$$w(M) \stackrel{\text{def}}{=} \sum_{e \in M} w(e)$$

Max. weight bipartite matching



Input:

bipartite graph $G = (U \sqcup V, E)$

weights $w : E \to \mathbb{R}_{\geq 0}$

Output:

matching $M \subseteq E$ maximizing

$$w(M) \stackrel{\text{def}}{=} \sum_{e \in M} w(e)$$

Approximate output:

matching $M \subseteq E$ s.t. $w(M) \geq (1 - \epsilon)\mathrm{OPT}$

Fast approximations: *bipartite matching*

	exact	approximate
cardinality	$O(m\sqrt{n})$ Hopcroft and Karp [1973] $\tilde{O}(m^{10/7})$ Mądry [2013]	$O(m/\epsilon)$
weighted	$O(mn + n^2 \log n)$ Fredman and Tarjan [1987] $O(m\sqrt{n} \log W)$ Duan and Su [2012]	$O(m \log(1/\epsilon)/\epsilon)$ Duan and Pettie [2014]

(m = edges, n = vertices, W = max weight)

Fast approximations: *bipartite matching*

	exact	approximate
cardinality	$O(m\sqrt{n})$ Hopcroft and Karp [1973] $\tilde{O}(m^{10/7})$ Mądry [2013]	$O(m/\epsilon)$
weighted	$O(mn + n^2 \log n)$ Fredman and Tarjan [1987] $O(m\sqrt{n} \log W)$ Duan and Su [2012]	$O(m \log(1/\epsilon)/\epsilon)$ Duan and Pettie [2014]

(m = edges, n = vertices, W = max weight)

(!) For fixed ϵ, weighted approximation is faster than unweighted exact

Duan and Pettie [2014]

(extends to general matching)

1. Primal-dual
   - *Only approximates dual optimal conditions*
2. Scaling reduces weighted problem to unweighted
3. Runs an approximate subroutine at each scale
4. Updates dual variables with small loss from approximation

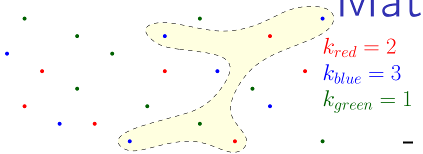
Matroids

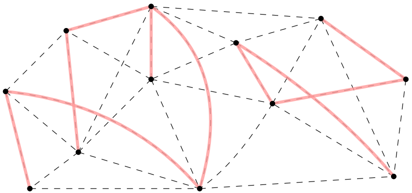
$$\mathcal{M} = (\mathcal{N}, \mathcal{I})$$

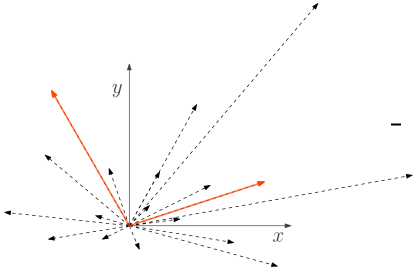
$\mathcal{N}$: **ground set** of elements

$\mathcal{I} \subseteq 2^{\mathcal{N}}$: **independent** (feasible) sets

Matroids







- Empty set is independent

- Subsets of independent sets are independent

- Maximal independent sets have same cardinality

max independent set is a ***base***

max cardinality is the ***rank***

- If $A, B \in \mathcal{I}$ and $|A| < |B|$, then there is $b \in B \setminus A$ s.t. $A + b \in \mathcal{I}$

Matroid Intersection

(same ground set)

$$\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1) \qquad \mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$$

$$\mathcal{M}_1 \cap \mathcal{M}_2 \triangleq (\mathcal{N}, \mathcal{I}_1 \cap \mathcal{I}_2)$$

e.g. bipartite matchings, arborescences

Matroid intersection problems

Maximum cardinality matroid intersection

Input: matroids $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$, $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$

Output: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $|S|$

Matroid intersection problems

Maximum cardinality matroid intersection

Input: matroids $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$, $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$

Output: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $|S|$

Maximum weight matroid intersection

Input: matroids $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$, $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$,
weights $w : \mathcal{N} \to \mathbb{R}_{\geq 0}$

Output: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $w(S) \stackrel{\text{def}}{=} \sum_{e \in S} w(s)$

Matroid intersection problems

Maximum cardinality matroid intersection

Input: matroids $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$, $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$

Output: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $|S|$

Maximum weight matroid intersection

Input: matroids $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$, $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$,
weights $w : \mathcal{N} \to \mathbb{R}_{\geq 0}$

Output: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $w(S) \stackrel{\text{def}}{=} \sum_{e \in S} w(s)$

Oracle Model

Independence queries of the form "Is $S \in \mathcal{I}$?"

Running times for matroid intersection

Running times for matroid intersection

Cardinality

$O(nk^{1.5}Q)$ Cunningham [1986]

($n = |\mathcal{N}|$, $k = \mathrm{rank}(\mathcal{M}_1 \cap \mathcal{M}_2)$, $Q =$ cost of indep. query)

Running times for matroid intersection

Cardinality

$O(nk^{1.5}Q)$ Cunningham [1986]

($n = |\mathcal{N}|$, $k = \mathrm{rank}(\mathcal{M}_1 \cap \mathcal{M}_2)$, $Q =$ *cost of indep. query*)

Weighted ($W = \max\{w(e) : e \in \mathcal{N}\}$)

$O(nk^2Q)$ Frank [1981], Brezovec et al. [1986], Schrijver [2003]

$O(n^2\sqrt{k}\log(kW)Q)$ Fujishige and Zhang [1995]

$O(nk^{1.5}WQ)$ Huang et al. [2014]

$O((n^2\log(n)Q + n^3\,\mathrm{polylog}(n))\log(nW))$

Lee et al. [2015]

Approximate matroid intersection

Input: $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$, $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$,
$w : \mathcal{N} \to \mathbb{R}_{\geq 0}$, $\epsilon > 0$

Output: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $w(S) \geq (1 - \epsilon)\mathrm{OPT}$

$(\mathrm{OPT} = \max\{w(T) : T \in \mathcal{I}_1 \cap \mathcal{I}_2\})$

Previous bound:

$O(nk^{1.5}\log(k)Q/\epsilon)$ Huang et al. [2014]

Approximate matroid intersection

Input: $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$, $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$,
$w : \mathcal{N} \to \mathbb{R}_{\geq 0}$, $\epsilon > 0$

Output: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $w(S) \geq (1-\epsilon)\mathrm{OPT}$
$(\mathrm{OPT} = \max\{w(T) : T \in \mathcal{I}_1 \cap \mathcal{I}_2\})$

Previous bound:

$O(nk^{1.5}\log(k)Q/\epsilon)$ Huang et al. [2014]

Main result: $(1-\epsilon)$-approximation in time

$$O(nkQ\log^2(\epsilon)/\epsilon^2)$$

Fast approximations: *matroid intersection*

	exact	approximate
cardinality	$O(nk^{1.5}Q)$ Cunningham [1986]	$O(nkQ/\epsilon)$
weighted	(nk^2Q) Frank [1981] and others $O(n^2\sqrt{k}\log(kW)Q)$ Fujishige and Zhang [1995] $O(nk^{1.5}WQ)$ Huang et al. [2014] $O((n^2\log(n)Q + n^3\operatorname{polylog}(n))\log(nW))$ Lee et al. [2015]	$O(nkQ\log^2(1/\epsilon)/\epsilon^2)$

(n = elements, k = rank, Q = indep. query, W = max weight)

Fast approximations: *matroid intersection*

	exact	approximate
cardinality	$O(nk^{1.5}Q)$ Cunningham [1986]	$O(nkQ/\epsilon)$
weighted	(nk^2Q) Frank [1981] and others $O(n^2\sqrt{k}\log(kW)Q)$ Fujishige and Zhang [1995] $O(nk^{1.5}WQ)$ Huang et al. [2014] $O((n^2\log(n)Q + n^3\,\mathrm{polylog}(n))\log(nW))$ Lee et al. [2015]	$O(nkQ\log^2(1/\epsilon)/\epsilon^2)$

(n = elements, k = rank, Q = indep. query, W = max weight)

(!) For fixed ϵ, weighted approximation is faster than unweighted exact approximate cardinality + approximate scaling

$\Rightarrow$ approximate solution

Some terminology

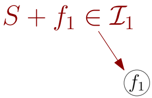
Matroid $\mathcal{M} = (\mathcal{N}, \mathcal{I})$, $S \in \mathcal{I}$

- **Span**: $e \in \mathrm{span}(S) \Rightarrow S + e \notin \mathcal{I}$

- **Free**: $e \in \mathrm{free}(S) \Rightarrow S + e \in \mathcal{I}$

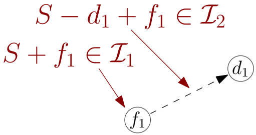
Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



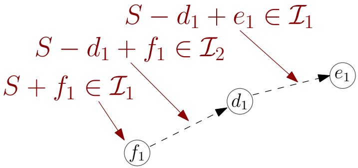
Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



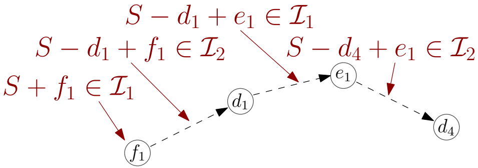
Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



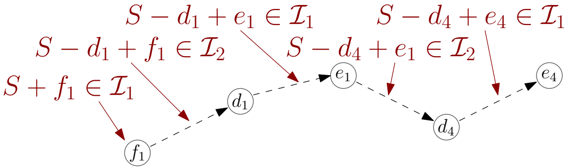
Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



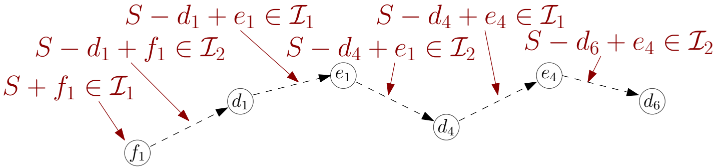
Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



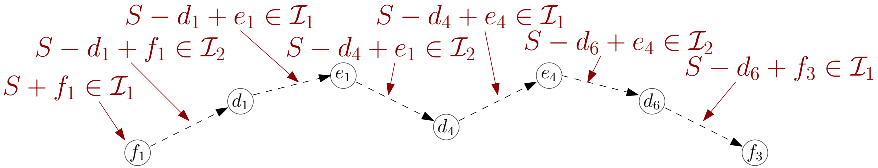
Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



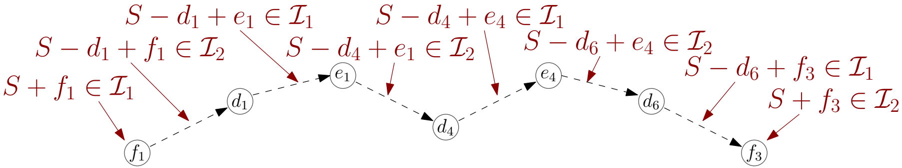
Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



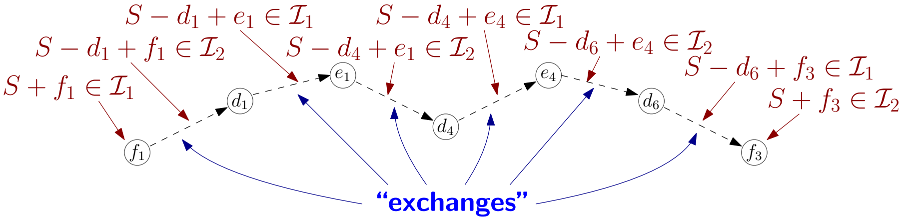
Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



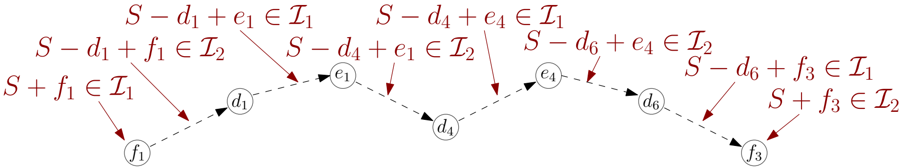
Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



Exchanges and exchange graphs

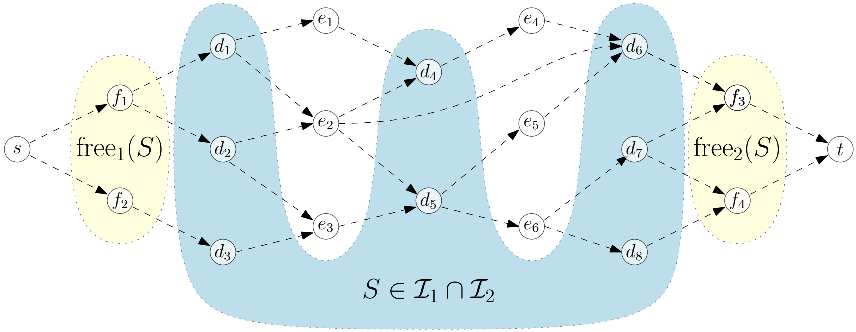
matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



"augmenting path" if $S + f_1 - d_1 + e_1 - d_4 + e_4 - d_6 + f_3 \in \mathcal{I}_1 \cap \mathcal{I}_2$

Exchanges and exchange graphs

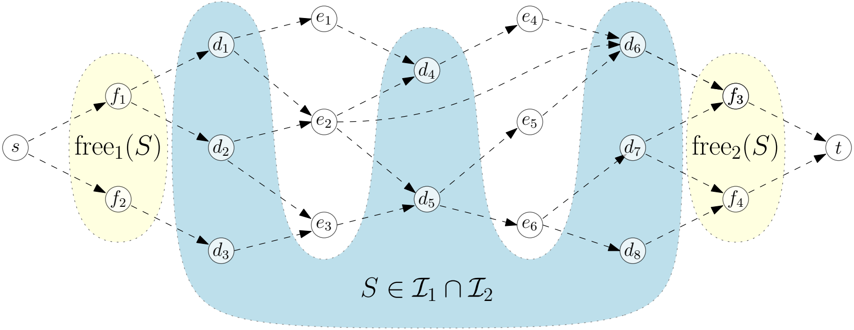
matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



exchange graph

Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

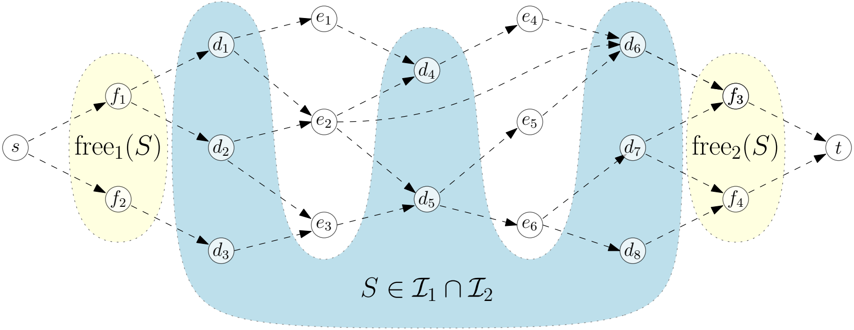


Bad news:

Good news:

Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

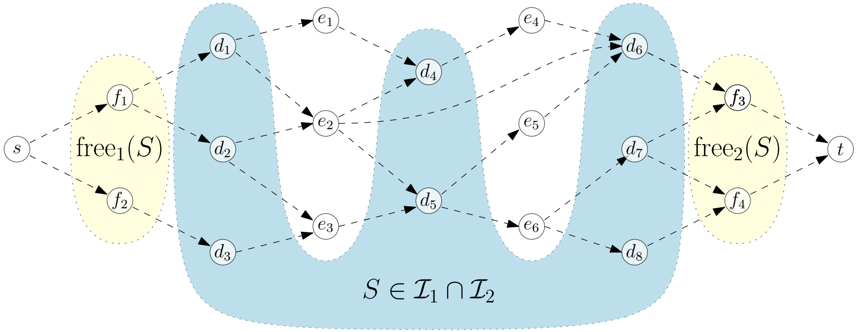


Bad news: not all paths are augmenting

Good news:

Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

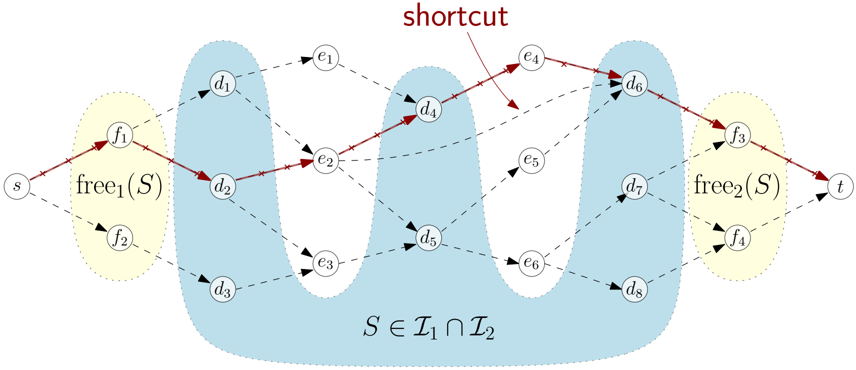


Bad news: not all paths are augmenting

Good news: shortest paths are augmenting

Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

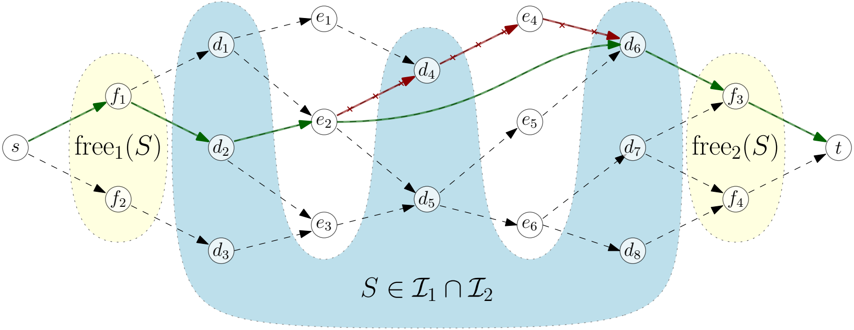


Bad news: not all paths are augmenting

Good news: shortest paths are augmenting

Exchanges and exchange graphs

matroid $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$, $S \in \mathcal{I}_1 \cap \mathcal{I}_2$



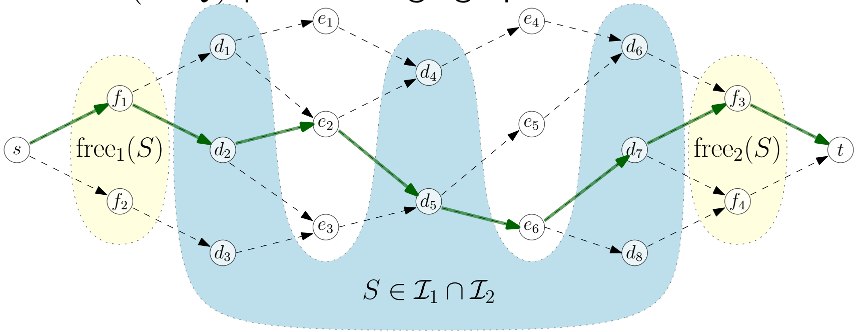
Bad news: not all paths are augmenting

Good news: shortest paths are augmenting

Unweighted matroid intersection

Naive alg.: augmenting paths $O(nk^2Q)$

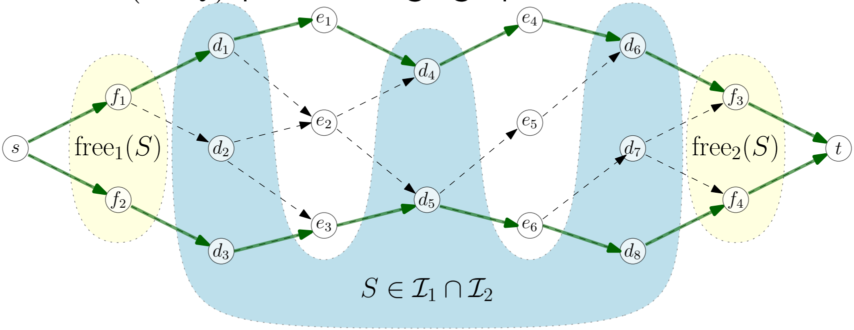
$O(nkQ)$ per exchange graph $\times$ k iterations



Unweighted matroid intersection

Naive alg.: augmenting paths $O(nk^2Q)$

$O(nkQ)$ per exchange graph $\times$ k iterations



Cunningham [1986]: $O(nk^{1.5}Q)$

Augment along several paths in each iteration

Cunningham's algorithm

Similar to Hopcroft-Karp for bipartite matching

Cunningham's algorithm

Similar to Hopcroft-Karp for bipartite matching

1. Increase length of short. aug. path in each iteration

subroutine: `batch-augment(`S`,`$\mathcal{M}_1$`,`$\mathcal{M}_2$`)`

input: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ with length of short. aug. path $= \ell$

output: $S' \in \mathcal{I}_1 \cap \mathcal{I}_2$ with length of short. aug. path $\geq \ell + 2$

running time: $O(nkQ)$

Cunningham's algorithm

Similar to Hopcroft-Karp for bipartite matching

1. Increase length of short. aug. path in each iteration

subroutine: `batch-augment(`S`,`$\mathcal{M}_1$`,`$\mathcal{M}_2$`)`
input: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ with length of short. aug. path $= \ell$
output: $S' \in \mathcal{I}_1 \cap \mathcal{I}_2$ with length of short. aug. path $\geq \ell + 2$
running time: $O(nkQ)$

2. Most augmenting paths are short

$\Rightarrow O(\sqrt{k})$ iterations

```
Cunningham(𝓜₁ = (𝓝, 𝓘₁), 𝓜₂ = (𝓝, 𝓘₂))
  S ← ∅
  repeat until S is fixed
    S ← batch-augment(S, 𝓜₁, 𝓜₂) // O(nkQ)
  // After √k iterations, |S| ≥ k − √k
  // O(√k) iterations total
  return S
```

```
Cunningham(M_1 = (N, I_1), M_2 = (N, I_2))
  S ← ∅
  repeat until S is fixed
    S ← batch-augment(S, M_1, M_2) // O(nkQ)
  // After √k iterations, |S| ≥ k − √k
  // O(√k) iterations total
  return S

Cunningham-APX(M_1, M_2)
  S ← ∅
  repeat O(1/ε) times
    S ← batch-augment(S, M_1, M_2) // O(nkQ)
  // |S| ≥ (1 − ε)k (!)
  return S // O(nkQ/ε)
```

Fast approximations: *matroid intersection*

	exact	approximate
cardinality	$O(nk^{1.5}Q)$ Cunningham [1986]	$O(nkQ/\epsilon)$
weighted	(nk^2Q) Frank [1981] and others $O(n^2\sqrt{k}\log(kW)Q)$ Fujishige and Zhang [1995] $O(nk^{1.5}WQ)$ Huang et al. [2014] $O((n^2\log(n)Q + n^3\,\mathrm{polylog}(n))\log(nW))$ Lee et al. [2015]	$O(nkQ\log^2(1/\epsilon)/\epsilon^2)$

(n = elements, k = rank, Q = indep. query, W = max weight)

Fast approximations: *matroid intersection*

	exact	approximate
cardinality	$O(nk^{1.5}Q)$ Cunningham [1986]	$O(nkQ/\epsilon)$
weighted	(nk^2Q) Frank [1981] and others $O(n^2\sqrt{k}\log(kW)Q)$ Fujishige and Zhang [1995] $O(nk^{1.5}WQ)$ Huang et al. [2014] $O((n^2\log(n)Q + n^3\operatorname{polylog}(n))\log(nW))$ Lee et al. [2015]	$O(nkQ\log^2(1/\epsilon)/\epsilon^2)$

(n = elements, k = rank, Q = indep. query, W = max weight)

Frank [1981]

Input: matroids $\mathcal{M}_1 = (\mathcal{N}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{N}, \mathcal{I}_2)$,
weights $w : \mathcal{N} \to \mathbb{R}_{\geq 0}$

Output: $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $w(S)$

Running time: $O(nk^2Q)$

1. Weight Splittings Frank [1981]

$$w_1, w_2 : \mathcal{N} \to \mathbb{R}_{\geq 0} \text{ s.t. } w = w_1 + w_2$$

Fact: *For* $S \in \mathcal{I}_1 \cap \mathcal{I}_2$*, if*

(a) $w_1(S) = \max_{\mathcal{I}_1} w_1(T)$*, and*

(b) $w_2(S) = \max_{\mathcal{I}_2} w_2(T)$

then $w(S) = \max\{w(T) : T \in \mathcal{I}_1 \cap \mathcal{I}_2\}$

2. Weight-induced matroid Frank [2008]

$$\mathcal{M}^w = (\mathcal{N}, \mathcal{I}^w)$$

$B \subseteq \mathcal{N}$ is a base in $\mathcal{M}^w$

$\iff$

B is a max. weight base in $\mathcal{M}$ w/r/t w

Reduces weighted matroid problems to unweighted problems

2. Weight-induced matroid Frank [2008]

$$\mathcal{M}^w = (\mathcal{N}, \mathcal{I}^w)$$

$B \subseteq \mathcal{N}$ is a base in $\mathcal{M}^w$

$\Longleftrightarrow$

B is a max. weight base in $\mathcal{M}$ w/r/t w

Reduces weighted matroid problems to unweighted problems

Frank's algorithm maintains weight splitting $w_1 + w_2 = w$ and $S \in \mathcal{I}_1^{w_1} \cap \mathcal{I}_2^{w_2}$.

```
Frank(M_1, M_2, w : N → R_{≥0})
    S ← ∅,  w_1 ← w,  w_2 ← 0
    // w_2(f) = 0 for all f ∈ free_2(S)




    // w_1(f) = 0 for all f ∈ free_1(S)
    // w_2(f) = 0 for all f ∈ free_2(S)
    return S
```

```
Frank(M_1, M_2, w : N → R_{≥0})
  S ← ∅,  w_1 ← w,  w_2 ← 0
  // w_2(f) = 0 for all f ∈ free_2(S)
  let c denote max{w_1(f) : f ∈ free_1(S)}
  while c > 0
    // free_1(f) ≤ c for all f ∈ free_1(S)






  end while
  // w_1(f) = 0 for all f ∈ free_1(S)
  // w_2(f) = 0 for all f ∈ free_2(S)
  return S
```

```
Frank(M_1, M_2, w : N → R_{≥0})
  S ← ∅, w_1 ← w, w_2 ← 0
  // w_2(f) = 0 for all f ∈ free_2(S)
  let c denote max{w_1(f) : f ∈ free_1(S)}
  while c > 0
    // free_1(f) ≤ c for all f ∈ free_1(S)
    N_c = {e : w(e) ≥ c}
    augment S in (M_1^{w_1} ∩ M_2^{w_2})|_{N_c} until optimal
    shift weight from w_1 to w_2

  end while
  // w_1(f) = 0 for all f ∈ free_1(S)
  // w_2(f) = 0 for all f ∈ free_2(S)
  return S
```

$$O(nk^2Q)$$

```
HKK(M_1, M_2, w : N → R_{≥0})
  S ← ∅,  w_1 ← w,  w_2 ← 0
  // w_2(f) = 0 for all f ∈ free_2(S)
  let c denote max{w_1(f) : f ∈ free_1(S)}
  for c = W, W − 1, ..., 3, 2, 1
    // free_1(f) ≤ c for all f ∈ free_1(S)
    N_c = {e : w(e) ≥ c}
    run Cunningham on S in (M_1^{w_1} ∩ M_2^{w_2})|_{N_c}
    shift one unit of weight from w_1 to w_2

  end for
  // w_1(f) = 0 for all f ∈ free_1(S)
  // w_2(f) = 0 for all f ∈ free_2(S)
  return S
```

$$O(nk^{1.5}WQ)$$

approximate cardinality $+$ approximate scaling

$=$ approximate weight splitting

$\Rightarrow$ approximate solution

```
HKK(M_1, M_2, w : N → R_{≥0})
  S ← ∅,  w_1 ← w,  w_2 ← 0
  // w_2(f) = 0 for all f ∈ free_2(S)
  let c denote max{w_1(f) : f ∈ free_1(S)}
  for c = W, W − 1, . . . , 3, 2, 1
    // free_1(f) ≤ c for all f ∈ free_1(S)
    N_c = {e : w(e) ≥ c}
    run Cunningham on S in (M_1^{w_1} ∩ M_2^{w_2})|_{N_c}
    shift one unit of weight from w_1 to w_2

  end for
  // w_1(f) = 0 for all f ∈ free_1(S)
  // w_2(f) = 0 for all f ∈ free_2(S)
  return S
```

$$O(nk^{1.5}WQ)$$

```
Frank-APX(M_1, M_2, w : N → ℝ_{≥0}, ε > 0)
  S ← ∅,  w_1 ← w,  w_2 ← 0
  // w_2(f) = 0 for all f ∈ free_2(S)
  let c denote max{w_1(f) : f ∈ free_1(S)}
  for c = W, W/2, ..., 4ε, 2ε, ε
    // free_1(f) ≤ c for all f ∈ free_1(S)
    N_c = {e : w(e) ≥ c}
    repeat 1/ε times:
      run Cunningham-APX on S in (M_1^{w_1} ∩ M_2^{w_2})|_{N_c}
      shift εc units of weight from w_1 to w_2
  end for
  // w_1(f) = 0 for all f ∈ free_1(S)
  // w_2(f) = 0 for all f ∈ free_2(S)
  return S
```

$$O(nkQ\log^2(1/\epsilon)/\epsilon^2)$$

approximate cardinality + approximate scaling

= approximate weight splitting

⇒ approximate solution

thank you

$$\boxed{\text{approximate cardinality}} + \boxed{\text{approximate scaling}}$$

$$= \boxed{\text{approximate weight splitting}}$$

$$\Rightarrow \boxed{\text{approximate solution}}$$

Bibliography I

C. Brezovec, G. Cornuéjols, and F. Glover. Two algorithms for weighted matroid intersection. *Math. Prog.*, 36(1):39–53, 1986.

W. H. Cunningham. Improved bounds for matroid partition and intersection algorithms. *SIAM J. Comput.*, 15(4):948–957, 1986.

R. Duan and S. Pettie. Linear-time approximation for maximum weight matching. *J. Assoc. Comput. Mach.*, 61(1), January 2014.

R. Duan and H. Su. A scaling algorithm for maximum weight matching in bipartite graphs. In *Proc. 23rd ACM-SIAM Sympos. Discrete Algs.* (SODA), pages 1413–1424, 2012.

A. Frank. A weighted matroid intersection algorithm. *J. Algorithms*, 2: 328–336, 1981.

A. Frank. A quick proof of the matroid intersection weight-splitting theorem. *EGRES Quick-Proofs*, 3, 2008.

M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. Assoc. Comput. Mach.*, 34(3):596–615, 1987.

Bibliography II

S. Fujishige and X. Zhang. An efficient cost scaling algorithm for the independent assignment problem. *J. Oper. Res. Soc. Japan*, 38(1): 124–136, 1995.

J. Hopcroft and R. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, pages 225–231, 1973.

C. Huang, N. Kakimura, and N. Kamiyama. Exact and approximation algorithms for weighted matroid intersection. *MI Preprint Series, Math. for Industry, Kyushu University*, November 2014. To appear in Proc. 27th ACM-SIAM Symp. Discrete Algs. *(SODA)*, 2016.

Y.T. Lee, A. Sidford, and S.C. Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proc. 56th Annu. IEEE Symp. Found. Comput. Sci.* (FOCS), 2015.

A. Mądry. Navigating central path with electrical flows: from flows to matchings, and back. In *Proc. 54th Annu. IEEE Symp. Found. Comput. Sci.* (FOCS), pages 253–262, 2013.

A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.