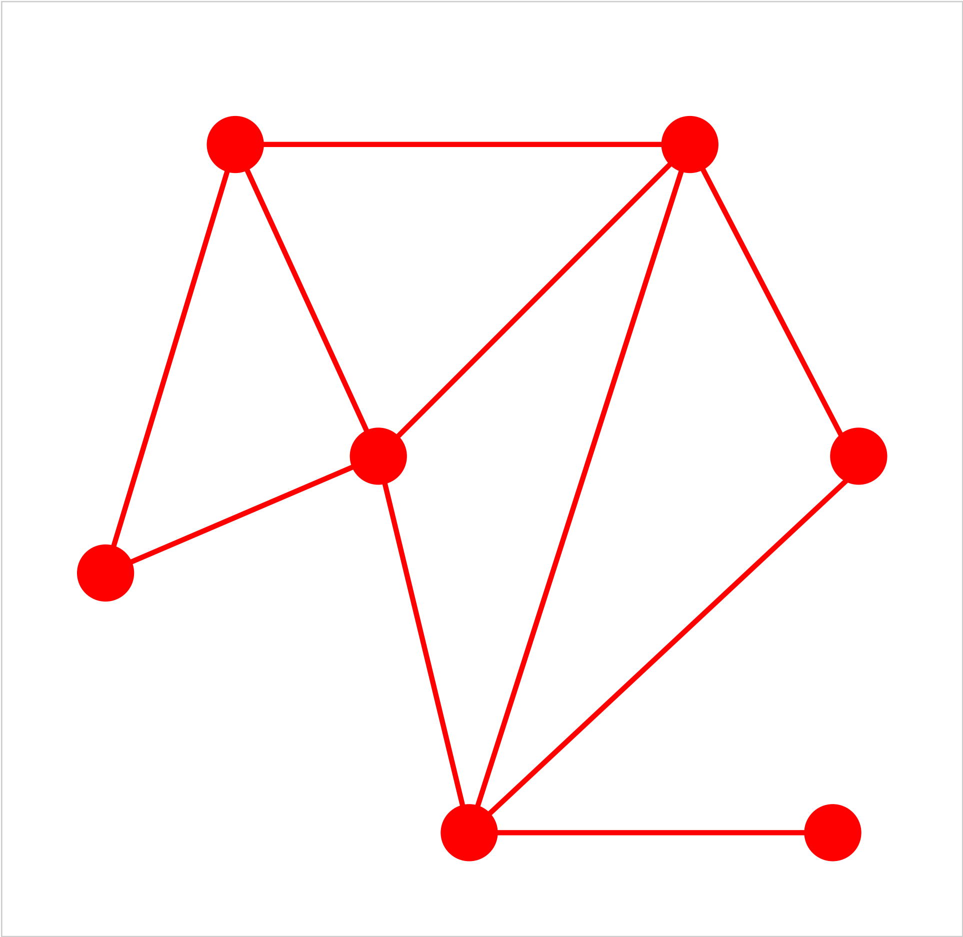
Submodular Maximization in a Data Streaming Setting

Amit Chakrabarti
Dartmouth College
Hanover, NH, USA

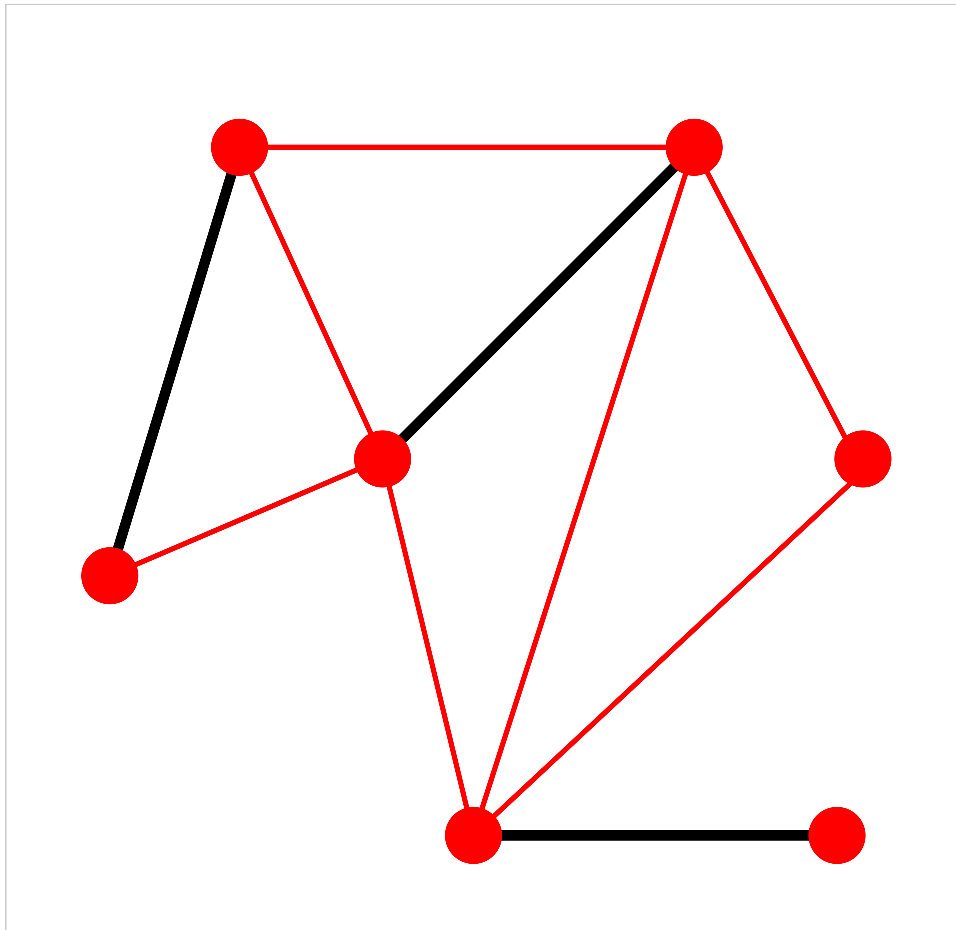
Based on joint work with Sagar Kale

Sublinear Algorithms Workshop
Bertinoro, May 2014

Maximum Matching

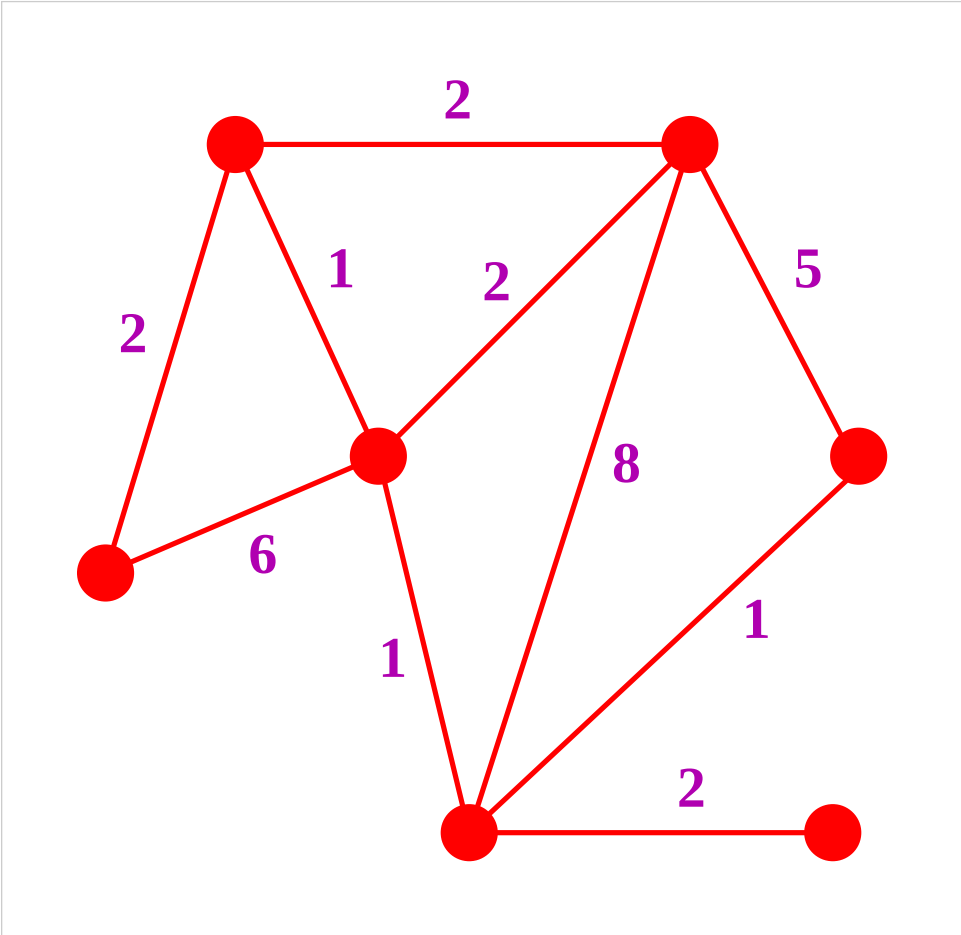


Maximum Matching

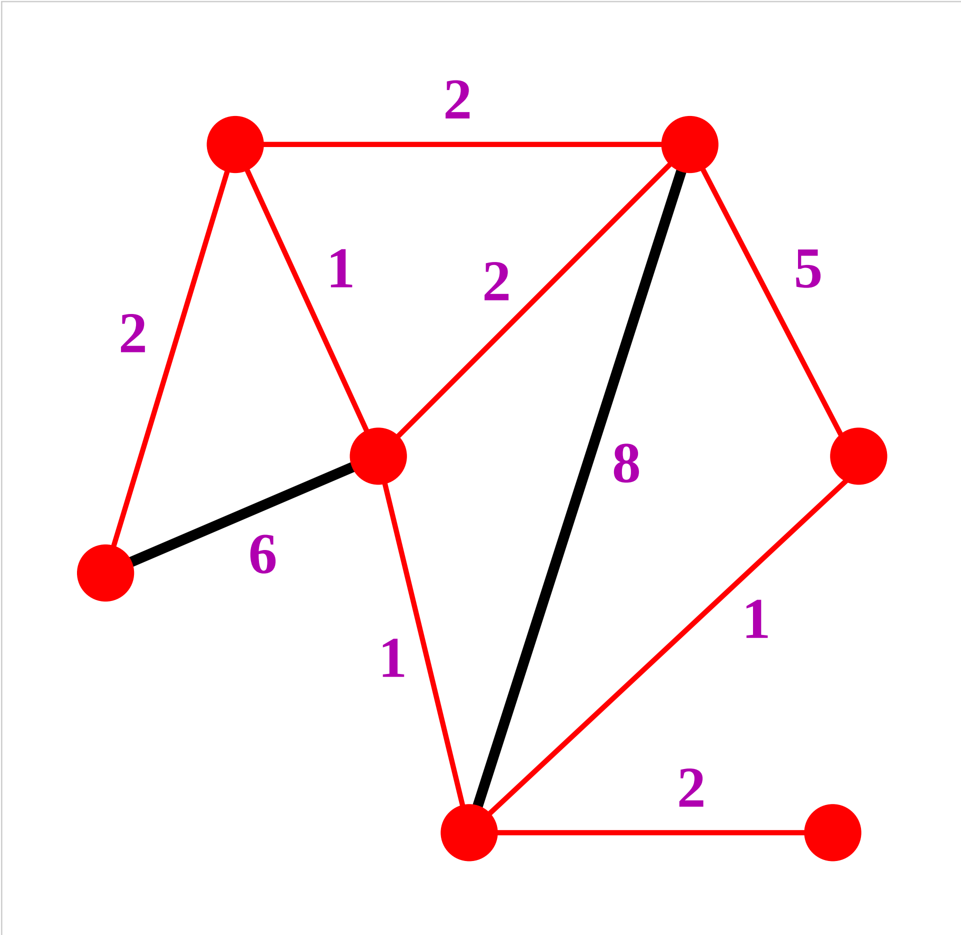


The cardinality version

Maximum Matching



Maximum Matching



The weighted version

Maximum Matching in a Graph Stream

Maximum cardinality matching (MCM)

- Input: stream of edges $(u, v) \in [n] \times [n]$
- Describes graph $G = (V, E)$: n vertices, m edges, undirected, simple
- Each edge appears exactly once in stream
- Goal
  - Output a matching $M \subseteq E$, with $|M|$ maximal

Maximum Matching in a Graph Stream

Maximum cardinality matching (MCM)

- Input: stream of edges $(u, v) \in [n] \times [n]$
- Describes graph $G = (V, E)$: n vertices, m edges, undirected, simple
- Each edge appears exactly once in stream
- Goal
  - Output a matching $M \subseteq E$, with $|M|$ maximal
  - Use sublinear (in m) working memory
  - Ideally $O(n \operatorname{polylog} n)$ ... "semi-streaming"
  - Need $\Omega(n \log n)$ to store M

Maximum Matching in a Graph Stream

Maximum cardinality matching (MCM)

- Input: stream of edges $(u, v) \in [n] \times [n]$
- Describes graph $G = (V, E)$: n vertices, m edges, undirected, simple
- Goal: output a matching $M \subseteq E$, with $|M|$ maximal

Maximum Matching in a Graph Stream

Maximum cardinality matching (MCM)

- Input: stream of edges $(u, v) \in [n] \times [n]$
- Describes graph $G = (V, E)$: n vertices, m edges, undirected, simple
- Goal: output a matching $M \subseteq E$, with $|M|$ maximal

Maximum weight matching (MWM)

- Input: stream of weighted edges $(u, v, w_{uv}) \in [n] \times [n] \times \mathbb{R}^+$
- Goal: output matching $M \subseteq E$, with $w(M) = \sum_{e \in M} w(e)$ maximal

Maximum Matching in a Graph Stream

Maximum cardinality matching (MCM)

- Input: stream of edges $(u, v) \in [n] \times [n]$
- Describes graph $G = (V, E)$: n vertices, m edges, undirected, simple
- Goal: output a matching $M \subseteq E$, with $|M|$ maximal

Maximum weight matching (MWM)

- Input: stream of weighted edges $(u, v, w_{uv}) \in [n] \times [n] \times \mathbb{R}^+$
- Goal: output matching $M \subseteq E$, with $w(M) = \sum_{e \in M} w(e)$ maximal

Maximum submodular-function matching (MSM) $\leftarrow$ this talk

- Input: unweighted edges (u, v), plus submodular $f : 2^E \to \mathbb{R}^+$
- Goal: output matching $M \subseteq E$, with $f(M)$ maximal

Maximum Submodular Matching

Input

- Stream of edges $\sigma = \langle e_1, e_2, \ldots, e_m \rangle$
- Valuation function $f : 2^E \to \mathbb{R}^+$
  - Submodular, i.e., $\forall\, X \subseteq Y \subseteq E\ \forall\, e \in E$

$$f(X + e) - f(X) \geq f(Y + e) - f(Y)$$

  - Monotone, i.e., $X \subseteq Y \implies f(X) \leq f(Y)$
  - Normalized, i.e., $f(\varnothing) = 0$
- Oracle access to f: query at $X \subseteq E$, get $f(X)$
  - May only query at $X \subseteq$ (stream so far)

Goal

- Output matching $M \subseteq E$, with $f(M)$ ~~maximal~~ "large"
- Store $O(n)$ edges and f-values

Our Results

Can't solve MSM exactly

- MCM, approx $< e/(e-1) \implies$ space $\omega(n \operatorname{polylog} n)$ [Kapralov'13]
- Offline MSM, approx $< e/(e-1) \implies n^{\omega(1)}$ oracle calls [this work]
  - Via cardinality-constrained submodular max [Nemhauser-Wolsey'78]

Our Results

Can't solve MSM exactly

- MCM, approx $< e/(e-1) \implies$ space $\omega(n \operatorname{polylog} n)$ [Kapralov'13]
- Offline MSM, approx $< e/(e-1) \implies n^{\omega(1)}$ oracle calls [this work]
  - Via cardinality-constrained submodular max [Nemhauser-Wolsey'78]

Our results, using $O(n)$ storage:

Theorem 1 MSM, one pass: 7.75-approx

Theorem 2 MSM, $(3+\varepsilon)$-approx in $O(e^{-3})$ passes

Our Results

Can't solve MSM exactly

- MCM, approx $< e/(e-1) \implies$ space $\omega(n \operatorname{polylog} n)$ [Kapralov'13]
- Offline MSM, approx $< e/(e-1) \implies n^{\omega(1)}$ oracle calls [this work]
  - Via cardinality-constrained submodular max [Nemhauser-Wolsey'78]

Our results, using $O(n)$ storage:

Theorem 1 MSM, one pass: 7.75-approx

Theorem 2 MSM, $(3 + \varepsilon)$-approx in $O(e^{-3})$ passes

More importantly:

Meta-Thm 1 Every compliant MWM approx alg $\rightarrow$ MSM approx alg

Our Results

Can't solve MSM exactly

- MCM, approx $< e/(e-1) \implies$ space $\omega(n \operatorname{polylog} n)$ [Kapralov'13]
- Offline MSM, approx $< e/(e-1) \implies n^{\omega(1)}$ oracle calls [this work]
  - Via cardinality-constrained submodular max [Nemhauser-Wolsey'78]

Our results, using $O(n)$ storage:

Theorem 1 MSM, one pass: 7.75-approx

Theorem 2 MSM, $(3+\varepsilon)$-approx in $O(e^{-3})$ passes

More importantly:

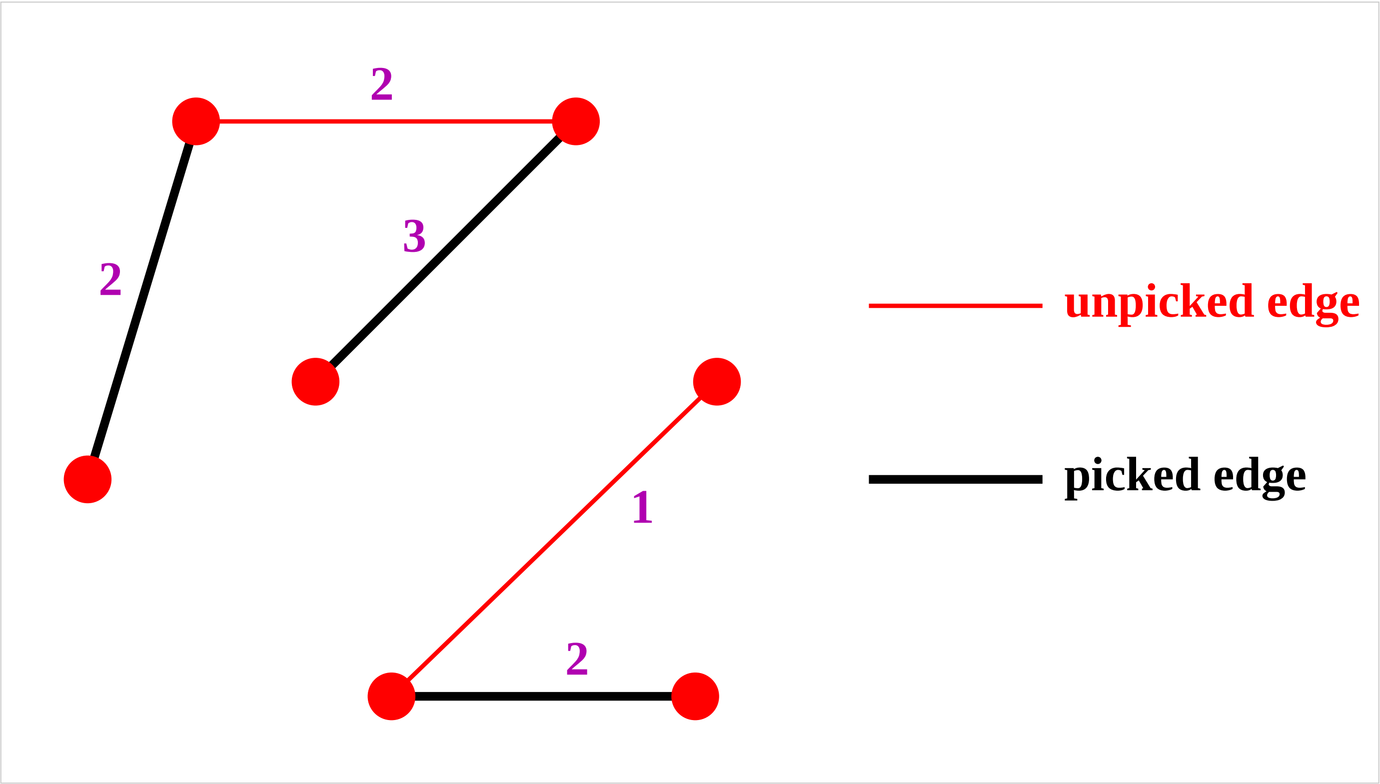
Meta-Thm 1 Every compliant MWM approx alg $\to$ MSM approx alg

Meta-Thm 2 Similarly, max weight independent set (MWIS) $\to$ MSIS

Some Previous Work on MWM

Greedy maximal matching: 2-approx for MCM, useless for MWM

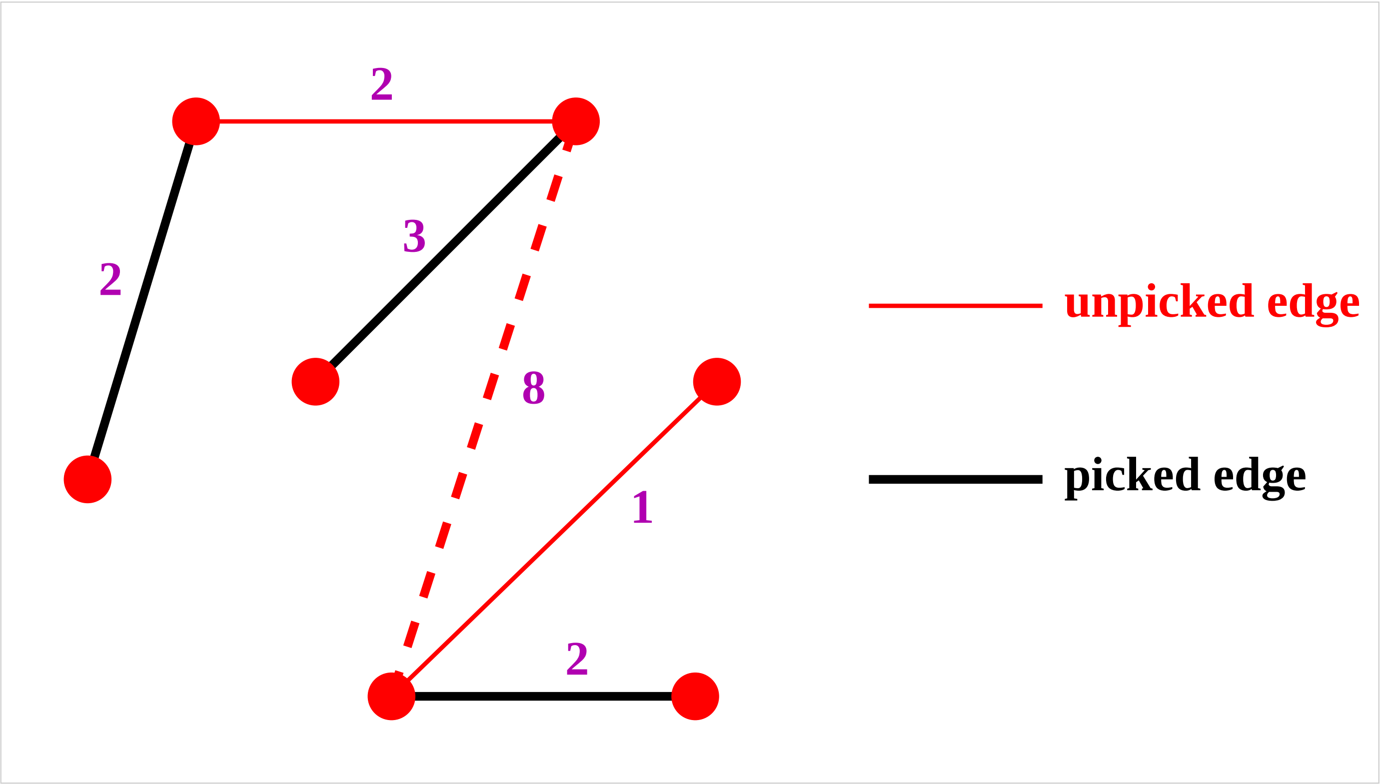
Maintain "current solution" M, update if new edge improves it



Some Previous Work on MWM

Greedy maximal matching: 2-approx for MCM, useless for MWM

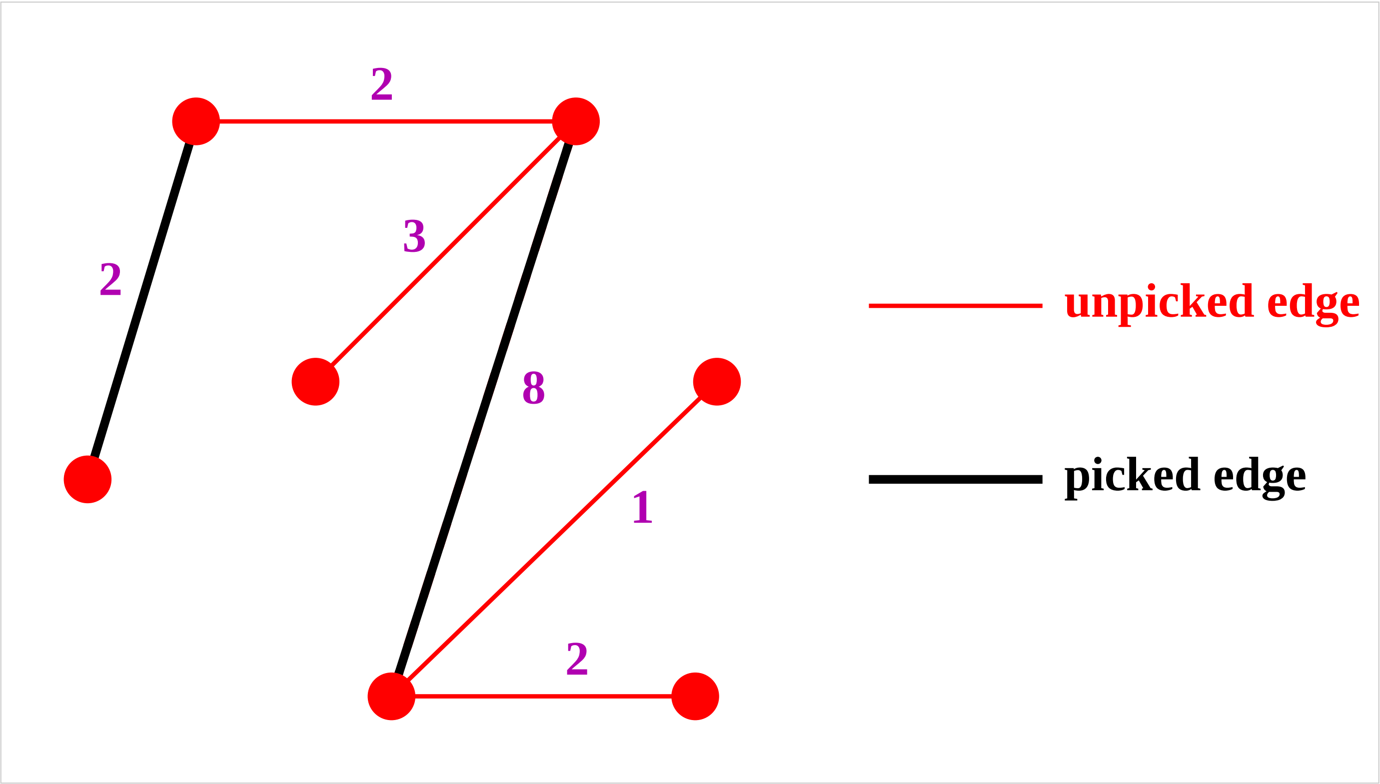
Maintain "current solution" M, update if new edge improves it



Some Previous Work on MWM

Greedy maximal matching: 2-approx for MCM, useless for MWM

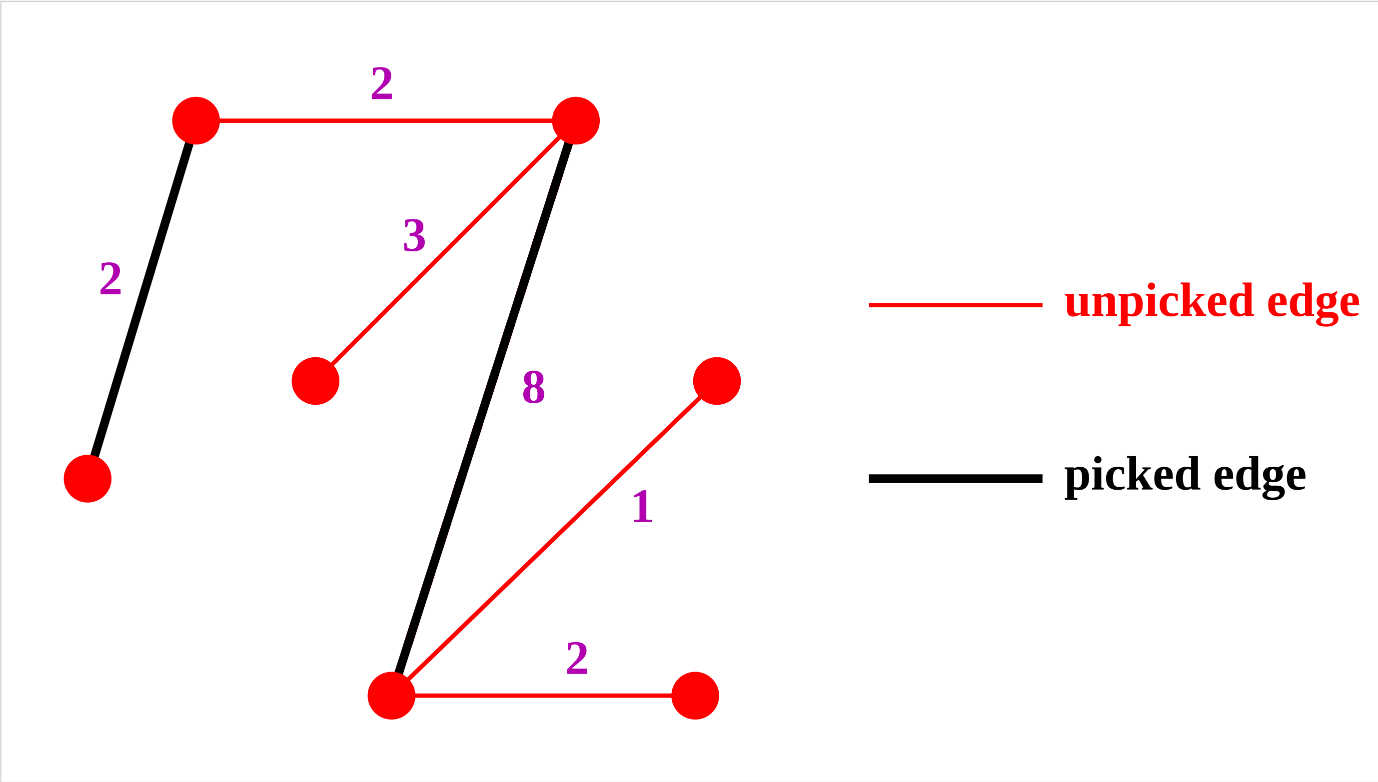
Maintain "current solution" M, update if new edge improves it



Compliant Algorithms for MWM

Greedy maximal matching: 2-approx for MCM, useless for MWM

Maintain "current solution" M, update if new edge improves it

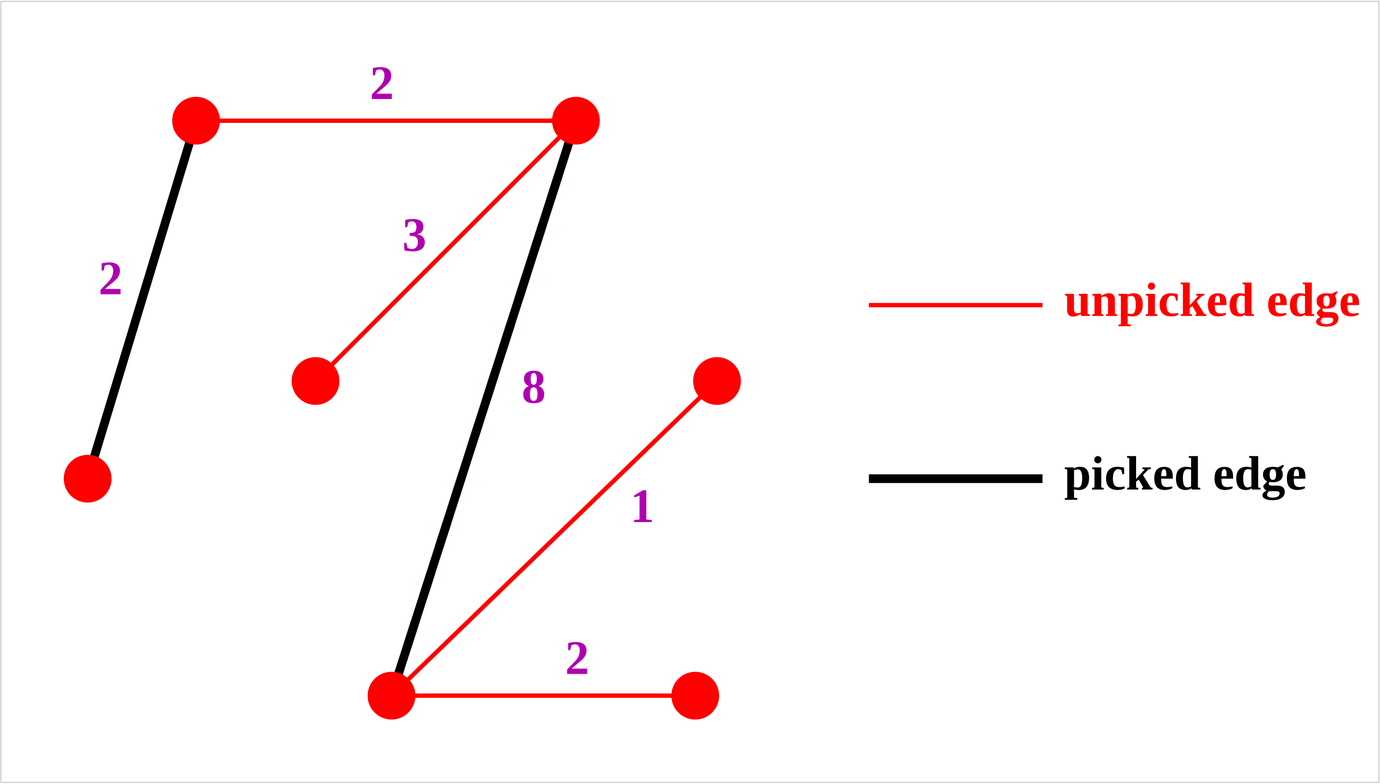


What if input is path with edge weights $1+\varepsilon, 1+2\varepsilon, 1+3\varepsilon, \ldots$?

Compliant Algorithms for MWM

Greedy maximal matching: 2-approx for MCM, useless for MWM

Maintain "current solution" M, update if new edge improves it



What if input is path with edge weights $1+\varepsilon, 1+2\varepsilon, 1+3\varepsilon, \ldots$?

Update M only upon <u>sufficient</u> improvement

Examples of Compliant Algorithms for MWM

Update of "current solution" M

- Given new edge e, pick "augmenting pair" (A, J)
  - $A \leftarrow \{e\}$
  - $J \leftarrow M \Cap A$ ... edges in M that conflict with A
  - Ensure $w(A) \geq (1+\gamma)w(J)$
- Update $M \leftarrow (M \setminus J) \cup A$

Examples of Compliant Algorithms for MWM

Update of "current solution" M

- Given new edge e, pick "augmenting pair" (A, J)
  - $A \leftarrow \{e\}$
  - $J \leftarrow M \Cap A$ ... edges in M that conflict with A
  - Ensure $w(A) \geq (1+\gamma)w(J)$
- Update $M \leftarrow (M \setminus J) \cup A$

Choice of gain parameter

- $\gamma = 1$, approx factor 6 [Feigenbaum-K-M-S-Z'05]
- $\gamma = 1/\sqrt{2}$, approx factor 5.828 [McGregor'05]

Examples of Compliant Algorithms for MWM

Update of "current solution" M

- Given new edge e, pick "augmenting pair" (A, J)
  - ~~$A \leftarrow \{e\}$~~ $A \leftarrow$ "best" subset of 3-neighbourhood of e
  - $J \leftarrow M \Cap A$ ... edges in M that conflict with A
  - Ensure $w(A) \geq (1 + \gamma)w(J)$
- Update $M \leftarrow (M \setminus J) \cup A$

Choice of gain parameter

- $\gamma = 1$, approx factor 6 [Feigenbaum-K-M-S-Z'05]
- $\gamma = 1/\sqrt{2}$, approx factor 5.828 [McGregor'05]
- $\gamma = 1.717$, approx factor 5.585 [Zelke'08]

Examples of Compliant Algorithms for MWM

Update of "current solution" M + pool of "shadow edges" S

- Given new edge e, pick "augmenting pair" (A, J)
  - ~~$A \leftarrow \{e\}$~~ $A \leftarrow$ "best" subset of 3-neighbourhood of e
  - $J \leftarrow M \Cap A$ ... edges in M that conflict with A
  - Ensure $w(A) \geq (1 + \gamma)w(J)$
- Update $M \leftarrow (M \setminus J) \cup A$
- Update $S \leftarrow$ appropriate subset of $(S \setminus A) \cup J$

Choice of gain parameter

- $\gamma = 1$, approx factor 6 [Feigenbaum-K-M-S-Z'05]
- $\gamma = 1/\sqrt{2}$, approx factor 5.828 [McGregor'05]
- $\gamma = 1.717$, approx factor 5.585 [Zelke'08]

Generic Compliant Algorithm and f-Extension for MSM

procedure PROCESS-EDGE(e, M, S, γ)

  $(A, J) \leftarrow$ a well-chosen augmenting pair for M
    with $A \subseteq M \cup S + e$, $w(A) \geq (1+\gamma)w(J)$

  $M \leftarrow (M \setminus J) \cup A$

  $S \leftarrow$ a well-chosen subset of $(S \setminus A) \cup J$

Generic Compliant Algorithm and f-Extension for MSM

procedure $\textsc{Process-Edge}(e, M, S, \gamma)$

$(A, J) \leftarrow$ a well-chosen augmenting pair for M
with $A \subseteq M \cup S + e$, $w(A) \geq (1+\gamma)w(J)$

$M \leftarrow (M \setminus J) \cup A$

$S \leftarrow$ a well-chosen subset of $(S \setminus A) \cup J$

MWM alg $\mathcal{A}$ + submodular f $\rightarrow$ MSM alg $\mathcal{A}^f$ (the f-extension of $\mathcal{A}$)

Generic Compliant Algorithm and f-Extension for MSM

procedure $\textsc{Process-Edge}(e, M, S, \gamma)$
    $w(e) \leftarrow f(M \cup S + e) - f(M \cup S)$
    $(A, J) \leftarrow$ a well-chosen augmenting pair for M
        with $A \subseteq M \cup S + e$, $w(A) \geq (1+\gamma)w(J)$
    $M \leftarrow (M \setminus J) \cup A$
    $S \leftarrow$ a well-chosen subset of $(S \setminus A) \cup J$

MWM alg $\mathcal{A}$ + submodular f $\rightarrow$ MSM alg $\mathcal{A}^f$ (the f-extension of $\mathcal{A}$)

Generic Compliant Algorithm and f-Extension for MSM

function IMPROVE-SOLUTION(σ, M^0, γ)
    $M \leftarrow \varnothing$, $S \leftarrow \varnothing$
    foreach $e \in M^0$ in arbit order **do** $w(e) \leftarrow f(M+e) - f(M)$, $M \leftarrow M + e$
    foreach $e \in \sigma \setminus M^0$ in the σ order **do** PROCESS-EDGE(e, M, S, γ)
    return M

procedure PROCESS-EDGE(e, M, S, γ)
    $w(e) \leftarrow f(M \cup S + e) - f(M \cup S)$
    $(A, J) \leftarrow$ a well-chosen augmenting pair for M
        with $A \subseteq M \cup S + e$, $w(A) \geq (1+\gamma)w(J)$
    $M \leftarrow (M \setminus J) \cup A$
    $S \leftarrow$ a well-chosen subset of $(S \setminus A) \cup J$

MWM alg $\mathcal{A}$ + submodular f $\rightarrow$ MSM alg $\mathcal{A}^f$ (the f-extension of $\mathcal{A}$)

Generic Compliant Algorithm and f-Extension for MSM

function $\textsc{Improve-Solution}(\sigma, M^0, \gamma)$
  $M \leftarrow \varnothing$, $S \leftarrow \varnothing$
  foreach $e \in M^0$ in arbit order **do** $w(e) \leftarrow f(M+e) - f(M)$, $M \leftarrow M + e$
  foreach $e \in \sigma \setminus M^0$ in the σ order **do** $\textsc{Process-Edge}(e, M, S, \gamma)$
  return M

procedure $\textsc{Process-Edge}(e, M, S, \gamma)$
  $w(e) \leftarrow f(M \cup S + e) - f(M \cup S)$
  $(A, J) \leftarrow$ a well-chosen augmenting pair for M
    with $A \subseteq M \cup S + e$, $w(A) \geq (1+\gamma)w(J)$
  $M \leftarrow (M \setminus J) \cup A$
  $S \leftarrow$ a well-chosen subset of $(S \setminus A) \cup J$

MWM alg $\mathcal{A}$ + submodular f $\rightarrow$ MSM alg $\mathcal{A}^f$ (the f-extension of $\mathcal{A}$)

MWIS (arbitrary ground set E, independent sets $\mathcal{I} \subseteq 2^E$) + f $\rightarrow$ MSIS

Generalize: Submodular Maximization (MWIS, MSIS)

function Improve-Solution(σ, I^0, γ)
 $I \leftarrow \varnothing$, $S \leftarrow \varnothing$
 foreach $e \in I^0$ in arbit order **do** $w(e) \leftarrow f(I+e) - f(I)$, $I \leftarrow I + e$
 foreach $e \in \sigma \setminus I^0$ in the σ order **do** Process-Element(e, I, S, γ)
 return I

procedure Process-Element(e, I, S, γ)
 $w(e) \leftarrow f(I \cup S + e) - f(I \cup S)$
 $(A, J) \leftarrow$ a well-chosen augmenting pair for I
 with $A \subseteq I \cup S + e$, $w(A) \geq (1+\gamma)w(J)$
 $I \leftarrow (I \setminus J) \cup A$
 $S \leftarrow$ a well-chosen subset of $(S \setminus A) \cup J$

MWIS (arbitrary ground set E, independent sets $\mathcal{I} \subseteq 2^E$) + f $\rightarrow$ MSIS

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \operatorname{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Lemma 1: Let $I_e, S_e =$ values of I, S just before e arrives

Then $I_e \cup S_e \supseteq \{x \in I^1 : x \prec e\} =: I^1_{\prec e}$ ("$\prec$": precedes in stream)

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \operatorname{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Lemma 1: Let $I_e, S_e =$ values of I, S just before e arrives

Then $I_e \cup S_e \supseteq \{x \in I^1 : x \prec e\} =: I^1_{\prec e}$ ("$\prec$": precedes in stream)

So, $f(I^1_{\preceq e}) - f(I^1_{\prec e}) \geq f(I_e \cup S_e + e) - f(I_e \cup S_e)$ (submodularity)

$= w(e)$ (definition of w)

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Lemma 1: Let $I_e, S_e =$ values of I, S just before e arrives

Then $I_e \cup S_e \supseteq \{x \in I^1 : x \prec e\} =: I^1_{\prec e}$ ("$\prec$": precedes in stream)

So, $f(I^1_{\preceq e}) - f(I^1_{\prec e}) \geq f(I_e \cup S_e + e) - f(I_e \cup S_e)$ (submodularity)

$= w(e)$ (definition of w)

Sum this over $x \in I^1$ in stream order, telescope *QED*

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\Longrightarrow$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \operatorname{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Lemma 2: Let $(A_e, J_e) =$ augmenting pair chosen on reading e

Then $w(A_e) \geq (1+\gamma) w(J_e)$

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Lemma 2: Let $(A_e, J_e) =$ augmenting pair chosen on reading e

Then $w(A_e) \geq (1+\gamma) w(J_e) \implies (w(A_e) - w(J_e))/\gamma \geq w(J_e)$

$\implies w(I^1)/\gamma \geq \sum_e w(J_e)$ (sum up)

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Lemma 2: Let $(A_e, J_e) =$ augmenting pair chosen on reading e

Then $w(A_e) \geq (1+\gamma) w(J_e) \implies (w(A_e) - w(J_e))/\gamma \geq w(J_e)$

$\implies w(I^1)/\gamma \geq \sum_e w(J_e)$ (sum up)

Each element in K was removed at some point

So, $K \subseteq \bigcup_e J_e \implies w(K) \leq \sum_e w(J_e) \leq w(I^1)/\gamma$ *QED*

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Lemma 3: Similar in spirit: submodularity, telescoping sums...

But a more involved argument

Uses Lemma 1 and Lemma 2

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Lemma 3: Similar in spirit: submodularity, telescoping sums...

But a more involved argument

Uses Lemma 1 and Lemma 2 *... uh, QED?*

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1)f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e$ was added to $I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1) f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Conclusion: $\mathcal{A}$ gives C_γ-approx for MWIS, so $w(I^*) \leq C_\gamma w(I^1)$

Analysis of MWIS Algorithm (One Pass)

Let $I^* = \text{argmax}_{I \in \mathcal{I}} f(I)$, $I^1 =$ output at end of pass

Let $K = \{e \in E : e \text{ was added to } I\} \setminus I^1$

Lemma 1 $w(I^1) \leq f(I^1)$

Lemma 2 $w(K) \leq w(I^1)/\gamma$

Lemma 3 $f(I^*) \leq (1/\gamma + 1)f(I^1) + w(I^*)$

Conclusion $\mathcal{A}$ is C_γ-approx $\implies$ $\mathcal{A}^f$ is $(C_\gamma + 1 + 1/\gamma)$-approx

---

Proof of Conclusion: $\mathcal{A}$ gives C_γ-approx for MWIS, so $w(I^*) \leq C_\gamma w(I^1)$

So, $f(I^*) \leq (1/\gamma + 1)f(I^1) + C_\gamma w(I^1)$ (by Lemma 3)

$\leq (1/\gamma + 1 + C_\gamma)f(I^1)$ (by Lemma 1)

QED

Applications of the Paradigm

1. Zelke's compliant algorithm for MWM has $C_\gamma = 3 + 2\gamma + \frac{1}{\gamma} - \frac{\gamma}{(1+\gamma)^2}$
   Take f-extension, set $\gamma = 1$ (optimal), get 7.75-approx to MSM

Applications of the Paradigm

1. Zelke's compliant algorithm for MWM has $C_\gamma = 3 + 2\gamma + \frac{1}{\gamma} - \frac{\gamma}{(1+\gamma)^2}$

   Take f-extension, set $\gamma = 1$ (optimal), get 7.75-approx to MSM

2. McGregor gives multi-pass compliant MWM algorithm

   Take f-extension, set $\gamma = 1$ for first pass, $\gamma = \varepsilon/3$ for other passes

   Make passes until solution doesn't improve much

   Extend MWM analysis to MSM, get $(3 + \varepsilon)$-approx, $O(\varepsilon^{-3})$ passes

Applications of the Paradigm

1. Zelke's compliant algorithm for MWM has $C_\gamma = 3 + 2\gamma + \frac{1}{\gamma} - \frac{\gamma}{(1+\gamma)^2}$

   Take f-extension, set $\gamma = 1$ (optimal), get 7.75-approx to MSM

2. McGregor gives multi-pass compliant MWM algorithm

   Take f-extension, set $\gamma = 1$ for first pass, $\gamma = \varepsilon/3$ for other passes

   Make passes until solution doesn't improve much

   Extend MWM analysis to MSM, get $(3 + \varepsilon)$-approx, $O(\varepsilon^{-3})$ passes

How good is a $(3 + \varepsilon)$-approx for MSM?

- Offline greedy: grow $I \leftarrow I + e$, maximizing $f(I + e)$

  This gives 3-approx for MSM [Nemhauser-Wolsey'78]

Applications of the Paradigm

1. Zelke's compliant algorithm for MWM has $C_\gamma = 3 + 2\gamma + \frac{1}{\gamma} - \frac{\gamma}{(1+\gamma)^2}$

   Take f-extension, set $\gamma = 1$ (optimal), get 7.75-approx to MSM

2. McGregor gives multi-pass compliant MWM algorithm

   Take f-extension, set $\gamma = 1$ for first pass, $\gamma = \varepsilon/3$ for other passes

   Make passes until solution doesn't improve much

   Extend MWM analysis to MSM, get $(3+\varepsilon)$-approx, $O(\varepsilon^{-3})$ passes

How good is a $(3+\varepsilon)$-approx for MSM?

- Offline greedy: grow $I \leftarrow I + e$, maximizing $f(I+e)$

  This gives 3-approx for MSM [Nemhauser-Wolsey'78]

- Recently: more sophisticated local search

  Gives $(2+\varepsilon)$-approx for MSM [Feldman-Naor-Schwartz-Ward'11]

Further Applications: Hypermatchings

Stream of hyperedges $e_1, e_2, \ldots, e_m \subseteq [n]$, each $|e_i| \leq p$

Hypermatching $=$ subset of pairwise disjoint edges

Further Applications: Hypermatchings

Stream of hyperedges $e_1, e_2, \ldots, e_m \subseteq [n]$, each $|e_i| \leq p$

Hypermatching $=$ subset of pairwise disjoint edges

Multi-pass MSM algorithm (compliant)

- Augment using only current edge e
- Use $\gamma = 1$ for first pass, $\gamma = \varepsilon/(p+1)$ subsequently
- Make passes until solution doesn't improve much

Results

- $4p$-approx in one pass
- $(p + 1 + \varepsilon)$-approx in $O(\varepsilon^{-3})$ passes

Further Applications: Maximization Over Matroids

Stream of elements $e_1, e_2, \ldots, e_m$ from ground set E

Matroids $(E, \mathcal{I}_1), \ldots, (E, \mathcal{I}_p)$, given by circuit oracles:

$$\text{Given } A \subseteq E, \text{ returns } \begin{cases} \smile, & \text{if } A \in \mathcal{I}_i \\ \text{a circuit in } A, & \text{otherwise} \end{cases}$$

Independent sets, $\mathcal{I} = \bigcap_i \mathcal{I}_i$; size parameter $n = \max_{I \in \mathcal{I}} |I|$

Further Applications: Maximization Over Matroids

Stream of elements $e_1, e_2, \ldots, e_m$ from ground set E

Matroids $(E, \mathcal{I}_1), \ldots, (E, \mathcal{I}_p)$, given by circuit oracles:

$$\text{Given } A \subseteq E, \text{ returns } \begin{cases} ☺, & \text{if } A \in \mathcal{I}_i \\ \text{a circuit in } A, & \text{otherwise} \end{cases}$$

Independent sets, $\mathcal{I} = \bigcap_i \mathcal{I}_i$; size parameter $n = \max_{I \in \mathcal{I}} |I|$

Recent MWIS algorithm (compliant) [Varadaraja'11]

- Augment using only current element e
- Remove $J = \{x_1, \ldots, x_p\}$,
  where $x_i :=$ lightest element in circuit formed in ith matroid

Further Applications: Maximization Over Matroids

Stream of elements $e_1, e_2, \ldots, e_m$ from ground set E

Independent sets, $\mathcal{I} = \bigcap_i \mathcal{I}_i$; size parameter $n = \max_{I \in \mathcal{I}} |I|$

Recent MWIS algorithm (compliant) [Varadaraja'11]

- Augment using only current element e
- Remove $J = \{x_1, \ldots, x_p\}$,
  where $x_i :=$ lightest element in circuit formed in ith matroid

Further Applications: Maximization Over Matroids

Stream of elements $e_1, e_2, \ldots, e_m$ from ground set E

Independent sets, $\mathcal{I} = \bigcap_i \mathcal{I}_i$; size parameter $n = \max_{I \in \mathcal{I}} |I|$

Recent MWIS algorithm (compliant) [Varadaraja'11]

- Augment using only current element e
- Remove $J = \{x_1, \ldots, x_p\}$,
  where $x_i :=$ lightest element in circuit formed in ith matroid

Follow paradigm: use f-extension of above algorithm

Results, using $O(n)$ storage

- $4p$-approx in one pass
- $(p + 1 + \varepsilon)$-approx in $O(\varepsilon^{-3})$ passes *

---

* Multi-pass analysis only works for partition matroids

Conclusions

- Identified framework (compliant algorithms) capturing several semi-streaming algorithms for constrained maximization
- Using framework, extended algs from linear to submodular maximization
- Applied to (hyper)matchings, (intersection of) matroids
- Can smoothly interpolate approx factor between linear f and general submodular f via curvature of f

Conclusions

- Identified framework (compliant algorithms) capturing several semi-streaming algorithms for constrained maximization
- Using framework, extended algs from linear to submodular maximization
- Applied to (hyper)matchings, (intersection of) matroids
- Can smoothly interpolate approx factor between linear f and general submodular f via <u>curvature</u> of f

Open Problems

- Extend matroid multi-pass result beyond partition matroids
- Capture recent MWM algorithms that beat Zelke [Crouch-Stubbs'14]
- Lower bounds??? Is MSM harder to approximate than MWM?