Submodular Function Maximization in Parallel via the Multilinear Relaxation

Chandra Chekuri Kent Quanrud

UIUC

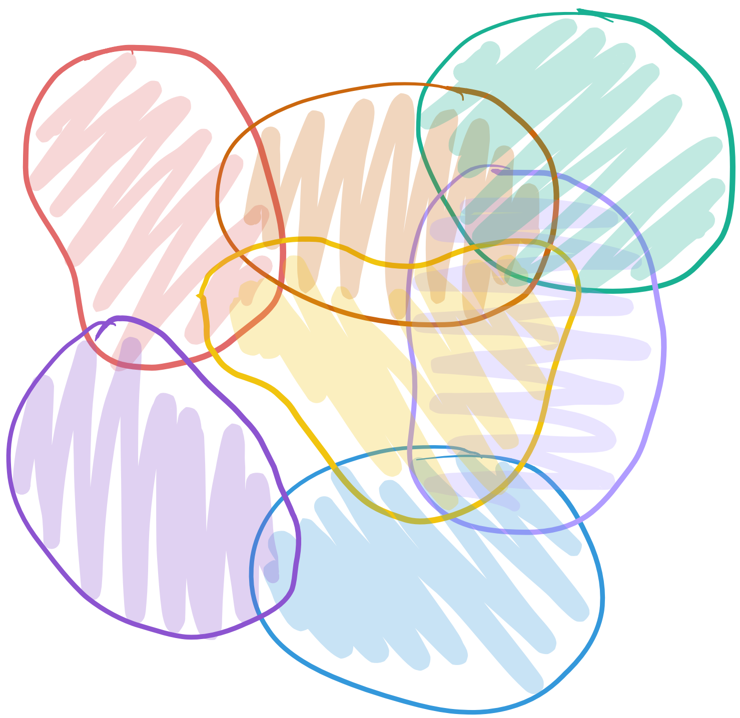
SODA 2019 January 6, 2019

e.g. Maximum coverage

Input: n sets $S_1, S_2, \ldots, S_n$, cardinality k

Goal: choose k sets $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$

maximizing their union $\left| \bigcup_{j=1}^{k} S_{i_j} \right|$



- NP-hard
- $1-\frac{1}{e}$ APX-hardness
- $1-\frac{1}{e}$ APX by greedy
- parallel?

GOAL:

maximize $f(S)$ s.t. $|S| \leq k$

in parallel in the oracle model

where

- $f : 2^N \to \mathbb{R}_{\geq 0}$ is a monotone submodular function
- $k \in \mathbb{N}$, alt: general packing constraints

Monotone submodular functions

set function $f: 2^N \to \mathbb{R}_{\geq 0}$

- $f(\emptyset) = 0$
- (increasing) $S \subseteq T \Rightarrow f(S) \leq f(T)$
- [decreasing marginal returns] denote $f_S(U) = f(S \cup U) - f(S)$

  = "marginal value of U to S"

  $$S \subseteq T \Rightarrow f_S(U) \geq f_T(U)$$

- Oracle: given S, returns $f(S)$

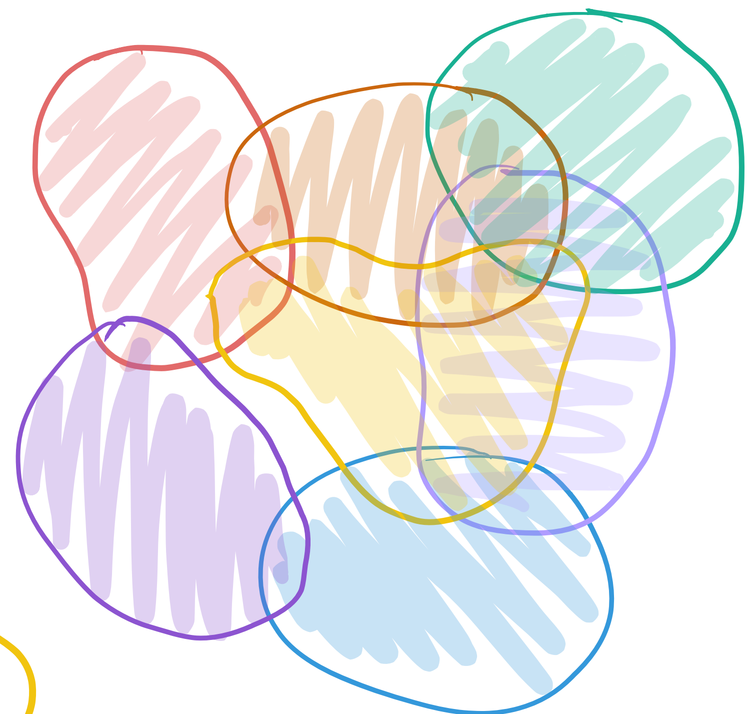
e.g. Maximum coverage

Input: n sets $S_1, S_2, \ldots, S_n$, cardinality k

we have a submodular function

$$f : 2^{N} \to \mathbb{R}_{\geq 0} \quad \text{where}$$

- $N = [n]$
- $f(I) = \left| \bigcup_{i \in I} S_i \right|$



(here "sets" get a little confusing)

Packing constraints

$$|S| \leq k \longrightarrow A \mathbb{1}_S \leq 1$$

$$A \in [0,1]^{m \times n}$$

e.g.

* Knapsack constraint

costs $a_1, a_2, \dots, a_n$ $\quad A = [a_1 \; a_2 \; \cdots \; a_n]$

* matchings
* partition matroids
* laminar matroids
* intersections of the above

Greedy algorithm [Nemhauser, Wolsey]

Input: $f : 2^N \to \mathbb{R}_{\geq 0}$, N, k

1. $S \leftarrow \emptyset$
2. while $|S| < k$
   a. $e \leftarrow \underset{e \in N}{\operatorname{argmax}}\, f_S(e)$ $\quad (= f(S+e) - f(S))$
   b. $S \leftarrow S + e$
3. return S

Greedy algorithm [Nemhauser, Wolsey]

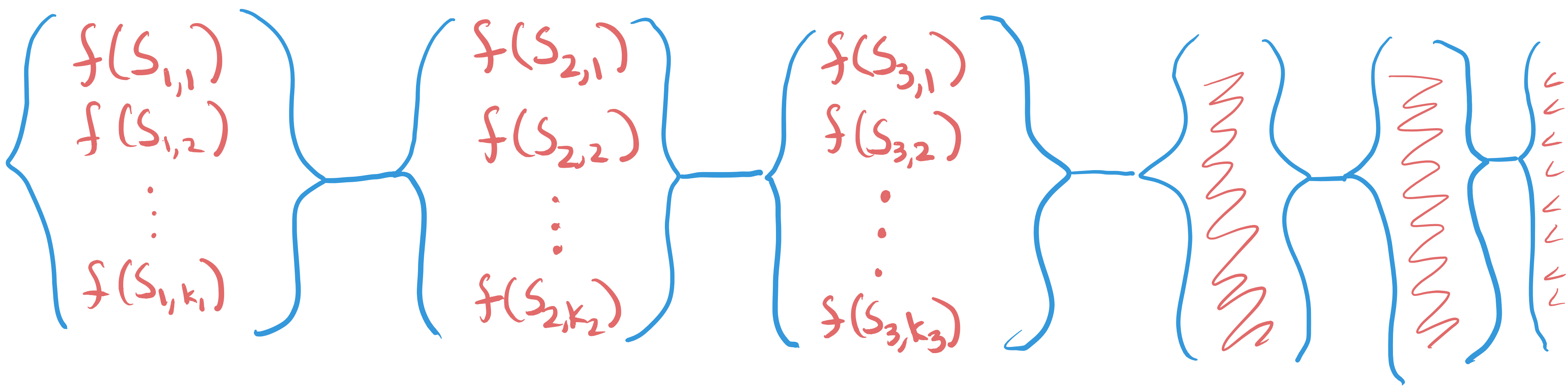
Input: $f: 2^N \to \mathbb{R}_{\geq 0}$, N, k

1. $S \leftarrow \emptyset$
2. while $|S| < k$
   a. $e \leftarrow \underset{e \in N}{\arg\max} f_S(e)$
   b. $S \leftarrow S + e$
3. return S

- simple
- Optimal $1 - \frac{1}{e}$ APX
- Sequential: reevaluates margins after each iteration

Adaptivity and parallelization

- Queries to f divided into "adaptive rounds"
- Choice of queries can only depend on queries to f of previous rounds



- e.g. greedy: nk total queries

  k adaptive rounds

Q: how much adaptivity/depth needed to maximize $f(S)$ s.t. $|S| \leq k$?

- greedy needs k

Parallel greedy???

Input: $f: 2^N \to \mathbb{R}_{\geq 0}$, N, k (large)

1. $S \leftarrow \emptyset$
2. while $|S| < k$

   /* need to take many (e.g. $\frac{k}{\text{polylog}(n)}$) elements.
   but elements negate each other */

   (e.g. overlapping sets in max coverage)

   a. $S \leftarrow S + ???$

3. return S

Q: how much adaptivity/depth needed to maximize $f(S)$ s.t. $|S| \le k$?

- greedy needs k
- [Balkanski & Singer] $\Omega\left(\frac{\log n}{\log\log n}\right)$ necessary

  $\frac{1}{3}$-APX w/ $O(\log n)$ rounds (randomized)
- [Balkanski, Rubinstein, Singer] [Ene, Nguyen] $(1-\frac{1}{e}-\varepsilon)$ w/ $O\left(\frac{\log n}{\text{poly}(\varepsilon)}\right)$ rounds (randomized)
- [Chekuri, Q.]: for general packing constraints

- [Balkanski, Rubinstein, Singer] [Ene, Nguyen] $(1-\frac{1}{e}-\varepsilon)$ w/ $O\left(\frac{\log n}{\mathrm{poly}(\varepsilon)}\right)$ rounds (randomized)
- [Chekuri, Q.]: for general packing constraints
  - different (simpler?) algorithm for cardinality constraints ("parallel-greedy") via multilinear extension of f
- deterministic for some explicit cases of f
- Cardinality ⇒ Knapsack ⇒ general packing via parallel multiplicative weight updates

multilinear extension of $f: 2^N \to \mathbb{R}_{\geq 0}$

$F: [0,1]^N \longrightarrow \mathbb{R}_{\geq 0}$ defined by

$F(x) = E[f(S)]$, where $\left(\begin{array}{l} S \text{ samples } e \in N \\ \text{independently with} \\ \text{probability } x_e \end{array} \right)$

- multilinear, monotone

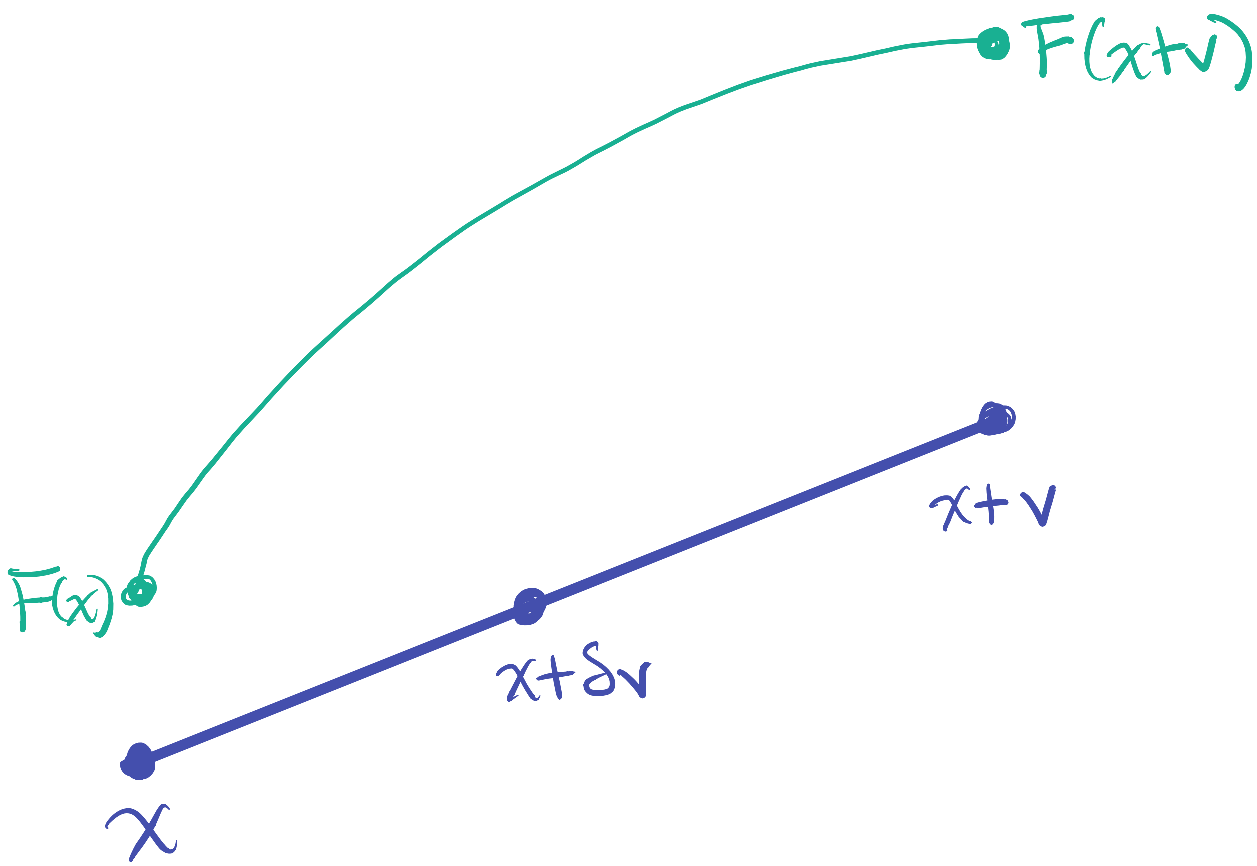
- monotone-concave: for $x \in [0,1]^N$, $v \in \mathbb{R}^N_{\geq 0}$, $\delta > 0$

  $F(x + \delta v)$ is concave in δ

- concentrated $\Rightarrow$ easy to estimate, round

- monotone-concave: for $x \in [0,1]^V$, $v \in \mathbb{R}^V_{\geq 0}$, $\delta > 0$

$F(x + \delta v)$ is concave in δ



multilinear extension of $f: 2^N \to \mathbb{R}_{\geq 0}$

$F: [0,1]^N \longrightarrow \mathbb{R}_{\geq 0}$ defined by

$F(x) = E[f(S)]$, where $\left\{ \begin{array}{l} S \text{ samples } e \in N \\ \text{independently with} \\ \text{probability } x_e \end{array} \right\}$

└ denote "$S \sim x$"

- multilinear, monotone
- monotone-concave: for $x \in [0,1]^N$, $v \in \mathbb{R}^N_{\geq 0}$, $\delta > 0$

  $F(x + \delta v)$ is concave in δ
- concentrated $\Rightarrow$ easy to estimate, round

Rounding (when $k \geq O(\frac{\log n}{\varepsilon^2})$)

given $x \in [0,1]^N$ w/ $\sum_{e \in N} x_e \leq k$,

let $y = (1-\varepsilon)x$

then for $S \sim y$

[$\text{Prob}[e \in S] = y_e$ independently $\forall e$]

- $|S| \leq k$ w/h/p
- $E[f(S)] = F(y) \overset{\star}{\geq} (1-\varepsilon) F(x)$

$\star$ by monotone concavity

multilinear extension of $f: 2^N \to \mathbb{R}_{\geq 0}$

$F: [0,1]^N \longrightarrow \mathbb{R}_{\geq 0}$ defined by

$F(x) = E[f(S)]$, where $\left\{ \begin{array}{l} S \text{ samples } e \in N \\ \text{independently with} \\ \text{probability } x_e \end{array} \right\}$

└ denote "$S \sim x$"

- multilinear, monotone
- monotone-concave: for $x \in [0,1]^N$, $v \in \mathbb{R}^N_{\geq 0}$, $\delta > 0$

  $F(x + \delta v)$ is concave in δ
- concentrated $\Rightarrow$ easy to estimate, round

GOAL:

maximize $f(S)$ s.t. $|S| \leq k$

in parallel in the oracle model

equivalent to

maximize $F(x)$ over $x \geq 0$ s.t. $\sum_i x_i \leq k$

in parallel in the oracle model

Continuous Greedy

[Wolsey] [Vondrák]
[Calinescu, Chekuri, Pál, Vondrák]

1. $x \leftarrow 0$
2. continuously from $t=0$ to k
   a. $v \leftarrow \underset{v \in P}{\operatorname{argmax}} \langle F'(x), v \rangle$

      where $P = \{ v \in [0,1]^N : \langle v, \mathbb{1} \rangle \leq 1 \}$
   b. $\frac{dx}{dt} = v$
3. return x

- $F'(x)$ = continuous analogue of marginal values
- greedily select fractional point v w/ maximum marginal value
- easy to discretize

- works for any downward closed polytope (e.g. matroids)

Continuous greedy in parallel

1. $x \leftarrow 0$
2. continuously from $t=0$ to k
   a. $\frac{dx}{dt} = \frac{S}{|S|}$, where $S = \{e : F'_e(x) \geq (1-\varepsilon) \max_d F'_d(x)\}$
3. return x

- S gathers all $(1-\varepsilon)$-OPT points in polytope
- increase x along all of S uniformly (instead of just the best point).

GOAL: discretize continuous algo to have low depth

Parallel greedy

1. $x \leftarrow \mathbb{0}$
2. while $\langle x, \mathbb{1} \rangle \leq k$
   a. $\lambda \leftarrow \max_e F_e'(x)$
   b. $S = \{ e \in N : F_e'(x) \geq (1-\varepsilon)\lambda \}$
   c. while $S \neq \emptyset$
      (i) $x \leftarrow x + \delta S$ for greedy step size $\delta > 0$
3. return x

- $S = \{$nearly best coordinates$\}$
- choose max δ s.t. $F(x+\delta S) - F(x) \geq (1-\varepsilon)^2 \delta |S| \lambda$

Parallel greedy

1. $x \leftarrow 0$
2. while $\langle x, \mathbb{1} \rangle \leq k$
   a. $\lambda \leftarrow \max_e F'_e(x)$
   b. $S = \{ e \in N : F'_e(x) \geq (1-\varepsilon)\lambda \}$
   c. while $S \neq \emptyset$
      (i) $x \leftarrow x + \delta S$ for greedy step size $\delta > 0$
3. return x

δ is:

- small enough to keep directional deriv $\langle F'(x+\delta S), S \rangle$ good, $\Rightarrow$ continuous greedy
- big enough to decrease directional deriv enough that ε-fraction of S drops out

"greedy step size" defined by

$$\max \delta$$
$$\text{s.t. } F(x+\delta S) - F(x) \geq (1-\varepsilon)^2 \delta |S| \lambda$$

$$\lambda \geq \max_e F'_e(x) \geq \frac{OPT - F(x)}{k}$$

$$\Rightarrow \frac{F(x+\delta S)-F(x)}{\delta |S|} = \frac{\text{"bang"}}{\text{"buck"}} \geq (1-\varepsilon)^2 \left(\frac{OPT-F(x)}{k}\right)$$

(a) $$\lim_{\delta \downarrow 0} \frac{F(x+\delta S)-F(x)}{\delta |S|} = \frac{\langle F'(x), S\rangle}{|S|} \geq (1-\varepsilon)\lambda$$

since $S = \{ i : F'_i(x) \geq (1-\varepsilon)\lambda$

(b) $\dfrac{F(x+\delta S)-F(x)}{\delta |S|}$ is decreasing in δ as $F'(x+\delta S)$ is decreasing "greedy step size" defined by

max δ

s.t. $F(x+\delta S) - F(x) \geq (1-\varepsilon)^2 \delta |S| \lambda$

$$\lambda \geq \max_e F'_e(x) \geq \frac{OPT - F(x)}{k}$$

$$\Rightarrow \frac{F(x+\delta S) - F(x)}{\delta |S|} = \frac{\text{"bang"}}{\text{"buck"}} \geq (1-\varepsilon)^2 \left(\frac{OPT - F(x)}{k} \right)$$

(a) $$\lim_{\delta \downarrow 0} \frac{F(x+\delta S) - F(x)}{\delta |S|} = \frac{\langle F'(x), S \rangle}{|S|} \geq (1-\varepsilon)\lambda$$

since $S = \{ i : F'_i(x) \geq (1-\varepsilon)\lambda$

(b) $\frac{F(x+\delta S) - F(x)}{\delta |S|}$ is decreasing in δ as $F'(x+\delta S)$ is decreasing

Analysis: (easy?)

① Approximation factor

(or how we choose δ small enough to approximately follow continuous greedy)

② Depth

(or how we choose δ big enough to kill off a large fraction of the good points S)

APX analysis

"greedy step size" δ is such that

$$\frac{F(x+\delta S)-F(x)}{\delta|S|} = \frac{\text{“bang”}}{\text{“buck”}} \geq (1-\varepsilon)^2\left(\frac{OPT-F(x)}{k}\right)$$

exact same dynamics as Greedy and cont. greedy

$$\Rightarrow \quad F(x) \geq (1-O(\varepsilon))(1-e^{-1})OPT$$

at the end, when $\sum x_i = k$

Analysis: (easy?)

1. Approximation factor

(or how we choose δ small enough to approximately follow continuous greedy)

2. Depth

(or how we choose δ big enough to kill off a large fraction of the good points S)

Depth

"greedy step size" is defined by

$$\max \delta \text{ s.t. } \frac{F(x+\delta S)-F(x)}{\delta|S|} = \frac{\text{"bang"}}{\text{"buck"}} \geq (1-\varepsilon)^2 \lambda$$

since is decreasing in δ and $\geq (1-\varepsilon)\lambda$ at $\delta=0$,

greedy step size $\delta \equiv$

$$\min \delta \text{ s.t. } \frac{F(x+\delta S)-F(x)}{\delta|S|} \leq (1-\varepsilon)^2 \lambda$$

* average margin of element in S has decreased

Depth

greedy step size $\delta \equiv$

$$\min \ \delta \ \text{ s.t. } \ \frac{F(x+\delta S)-F(x)}{\delta |S|} \le (1-\varepsilon)^2 \lambda$$

* average margin of element in S has decreased

Markov-type argument $\Rightarrow$

$|S|$ decreases by $(1-\varepsilon)$-mult. factor

$\Rightarrow \cdots \Rightarrow \cdots \Rightarrow$ $\text{poly}(\log n, \frac{1}{\varepsilon})$ depth

Analysis: (easy?)

① ✓ Approximation factor

(or how we choose δ small enough to approximately follow continuous greedy)

② ✓ Depth

(or how we choose δ big enough to kill off a large fraction of the good points S)

$\blacksquare$

Parallel greedy

1. $x \leftarrow \mathbb{0}$
2. while $\langle x, \mathbb{1} \rangle \le k$
   a. $\lambda \leftarrow \max_e F'_e(x)$
   b. $S = \{ e \in N : F'_e(x) \ge (1-\varepsilon)\lambda \}$
   c. while $S \neq \emptyset$
      (i) $x \leftarrow x + \delta S$ for greedy step size $\delta > 0$
3. return x

"primal-dual" approach

- continuous greedy (along S) ensures $(1-\frac{1}{e})$ APX
- greedy step size drives down margins and limits # iterations

Parallel greedy

1. $x \leftarrow \mathbb{0}$
2. while $\langle x, \mathbb{1} \rangle \leq k$
   a. $\lambda \leftarrow \max_e F'_e(x)$
   b. $S = \{ e \in \mathcal{N} : F'_e(x) \geq (1-\varepsilon) \lambda \}$
   c. while $S \neq \emptyset$
      (i) $x \leftarrow x + \delta S$ for greedy step size $\delta > 0$
3. return x

Q: is there an equally simple but combinatorial analog?

Randomized-Parallel-Greedy

("combinatorial version")

1. $Q \leftarrow \emptyset$,
2. while $|Q| \leq k$ and
   a. $\lambda \leftarrow \max_e F_e'(x)$
   b. $S = \{e \in N : f_Q(e) \geq (1-\varepsilon)\lambda/k\}$
   while $S \neq \emptyset$
   (i) $Q \leftarrow Q \cup R$ for R sampling δS w/r/t greedy step size $\delta > 0$
3. return Q

---

maintain discrete set Q by immediately rounding δS in each iteration

Beyond cardinality

- From continuous viewpoint,

  cardinality $\approx$ knapsack $\overset{1}{\approx}$ linear packing constraints

1 via MWU:

Integrate submodular techniques w/ "parallel MWU" techniques [Young '02]

- From continuous viewpoint,

cardinality $\approx$ knapsack $\approx$ linear packing constraints

* For knapsack, "good coordinates" defined by "bang-for-buck ratio" $\dfrac{F_e'(x)}{c_e}$ [where c_e = cost of item e]

* MWU uses Lagrangian relaxations to collapse to a sequence of knapsack problems.

Conclusion

- Recent interest in parallel submodular max: different constraints, nonnegative setting, better depth, better oracle complexity

[Balkanski Singer] [Ene Nguyen] [Balkanski Rubinstein Singer]

[Fahrbach Mirrokni Zadimoghaddam] [Ene Nguyen Vladu] [Chen Feldman Karbasi]

(w/ multiplicities)

Unifying techniques

- [continuous greedy]
  + [bulk update]
  + [greedy step size]

parallel greedy w/ low depth

- easily discretized
- extends to general packing via MWU

- matroids too by similar techniques

THANKS