Streaming Algorithm for Monotone k-Submodular Maximization with Cardinality Constraints

Alina Ene
Boston University

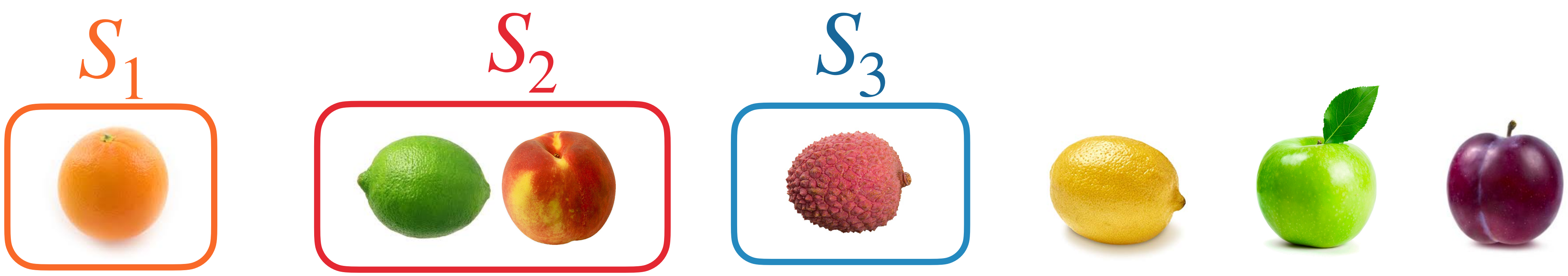
Huy L. Nguyen
Northeastern University

Problem Definition

V : finite set of n items

k : number of parts

$B_1, \ldots, B_k$: size constraint for each part



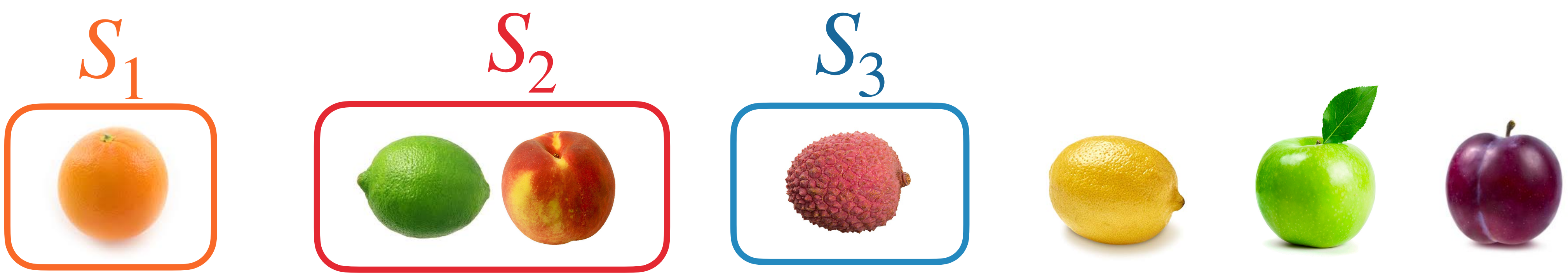
Goal: Select a k-tuple $\mathbf{S} = (S_1, S_2, \ldots, S_k)$ of <u>disjoint</u> subsets of V satisfying $|S_i| \leq B_i$ for all $i \in [k]$ that maximizes $f(\mathbf{S})$

Problem Definition

V : finite set of n items

k : number of parts

$B_1, \ldots, B_k$: size constraint for each part



Goal: Select a k-tuple $\mathbf{S} = (S_1, S_2, \ldots, S_k)$ of <u>disjoint</u> subsets of V satisfying $|S_i| \leq B_i$ for all $i \in [k]$ that maximizes $f(\mathbf{S})$

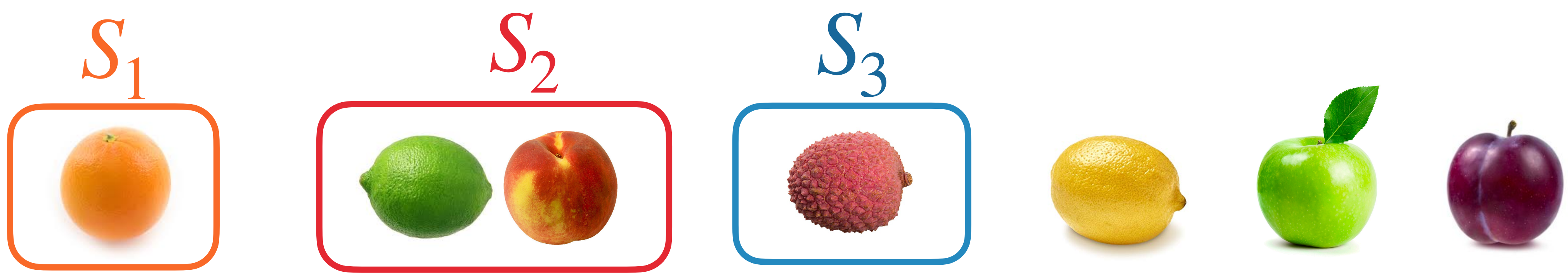
f is k-submodular and monotone

Problem Definition

V : finite set of n items

k : number of parts

$B_1, \ldots, B_k$: size constraint for each part



Goal: Select a k-tuple $\mathbf{S} = (S_1, S_2, \ldots, S_k)$ of <u>disjoint</u> subsets of V satisfying $|S_i| \leq B_i$ for all $i \in [k]$ that maximizes $f(\mathbf{S})$

f is k-submodular and monotone

$\Delta_{e,i} f(\mathbf{S}) \geq \Delta_{e,i} f(\mathbf{T})$ whenever $S_1 \subseteq T_1, \ldots, S_k \subseteq T_k$

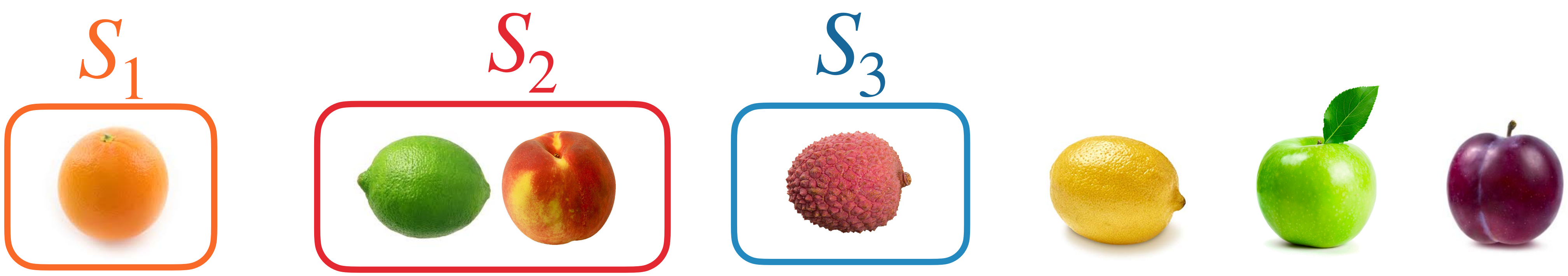
$\Delta_{e,i} f(\mathbf{S}) =$ marginal gain of adding e to S_i

Problem Definition

V : finite set of n items

k : number of parts

$B_1, \ldots, B_k$: size constraint for each part



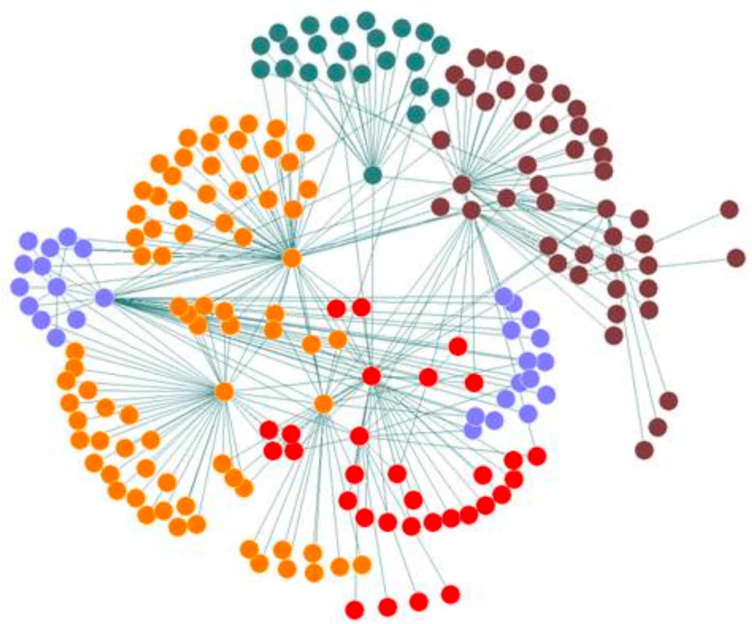
Goal: Select a k-tuple $\mathbf{S} = (S_1, S_2, \ldots, S_k)$ of <u>disjoint</u> subsets of V satisfying $|S_i| \leq B_i$ for all $i \in [k]$ that maximizes $f(\mathbf{S})$

f is k-submodular and monotone

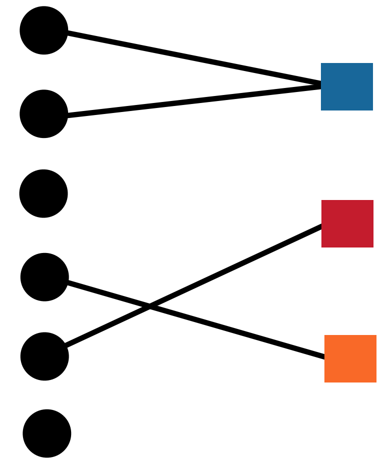
$f(\mathbf{S}) \leq f(\mathbf{T})$ whenever $S_1 \subseteq T_1, \ldots, S_k \subseteq T_k$

Motivating Applications

Influence maximization with k products

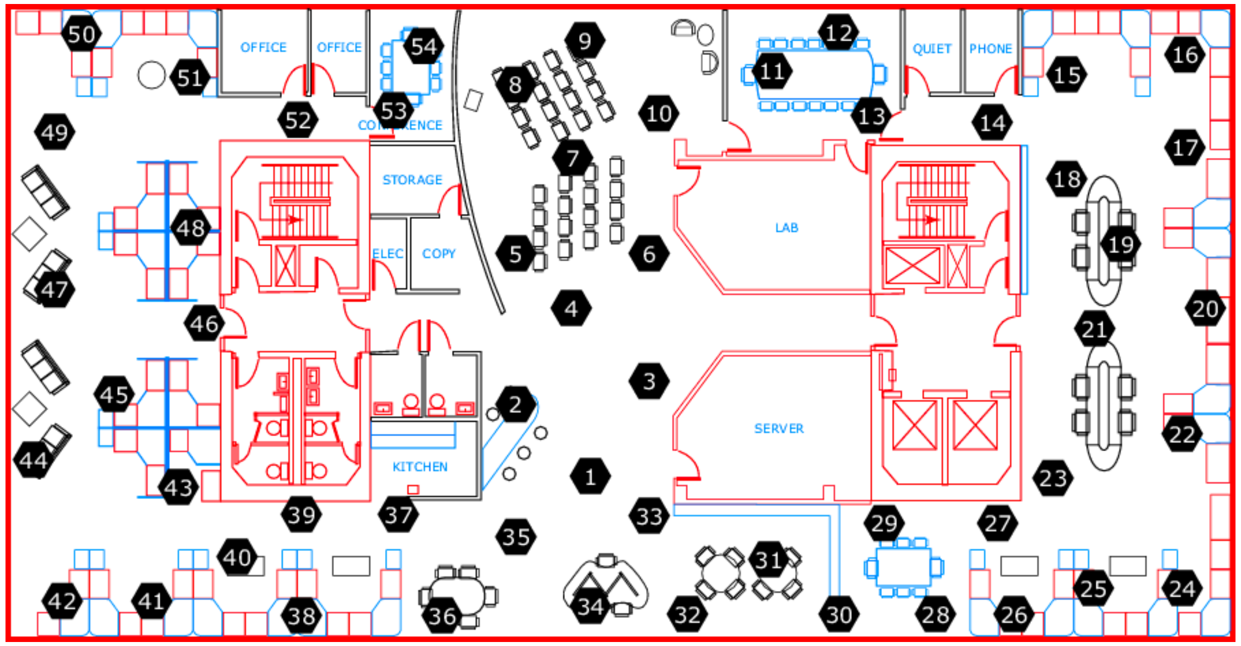


Online advertising



n impressions k advertisers

Sensor placement with k sensor types



Prior Work and Our Contributions

	Approx	Time	Space
Greedy [Ohsaka & Yoshida '15]	$\frac{1}{3}$	$O(nkr)$	$O(n)$

n = number of items $\quad k$ = number of parts $\quad B = \min_{i \in [k]} B_i$

$r = \sum_{i=1}^{k} B_i$ = total size of feasible solution

Prior Work and Our Contributions

	Approx	Time	Space
Greedy [Ohsaka & Yoshida '15]	$\frac{1}{3}$	$O(nkr)$	$O(n)$
Random Greedy [Ohsaka & Yoshida '15]	$\frac{1}{3}$	$\widetilde{O}(nk^2)$	$O(n)$

n = number of items k = number of parts $B = \min_{i \in [k]} B_i$

$r = \sum_{i=1}^{k} B_i$ = total size of feasible solution

Prior Work and Our Contributions

	Approx	Time	Space
Greedy [Ohsaka & Yoshida '15]	$\frac{1}{3}$	$O(nkr)$	$O(n)$
Random Greedy [Ohsaka & Yoshida '15]	$\frac{1}{3}$	$\widetilde{O}(nk^2)$	$O(n)$
Our algorithm (online, streaming)	$\frac{1}{2\left(1+B\left(2^{1/B}-1\right)\right)} \geq \frac{1}{4}$ $\to 0.2953$ as $B \to \infty$	$O(nk)$	$O(r)$

n = number of items k = number of parts $B = \min_{i \in [k]} B_i$

$r = \sum_{i=1}^{k} B_i$ = total size of feasible solution

Prior Work and Our Contributions

	Approx	Time	Space
Greedy [Ohsaka & Yoshida '15]	$\frac{1}{3}$	$O(nkr)$	$O(n)$
Random Greedy [Ohsaka & Yoshida '15]	$\frac{1}{3}$	$\widetilde{O}(nk^2)$	$O(n)$
Our algorithm (online, streaming)	$\frac{1}{2\left(1+B\left(2^{1/B}-1\right)\right)} \geq \frac{1}{4}$ $\rightarrow 0.2953$ as $B \rightarrow \infty$	optimal $O(nk)$	optimal $O(r)$

$n =$ number of items $k =$ number of parts $B = \min_{i \in [k]} B_i$

$r = \sum_{i=1}^{k} B_i =$ total size of feasible solution

Prior Work and Our Contributions

	Approx	Time	Space
Greedy [Ohsaka & Yoshida '15]	$\frac{1}{3}$	$O(nkr)$	$O(n)$
Random Greedy [Ohsaka & Yoshida '15]	$\frac{1}{3}$	$\widetilde{O}(nk^2)$	$O(n)$
Our algorithm (online, streaming)	$\frac{1}{2\left(1+B\left(2^{1/B}-1\right)\right)} \geq \frac{1}{4}$ $\to 0.2953$ as $B \to \infty$	optimal $O(nk)$	optimal $O(r)$

Our algorithm extends to submodular maximization with a partition matroid constraint

LP Formulation

$x_{e,i} = 1$ if item $e \in V$ is assigned to part $i \in [k]$

$y_{\mathbf{S}} = 1$ if $\mathbf{S} = (S_1, \ldots, S_k)$ is the k-tuple we select

Primal LP

$$\max \sum_{\mathbf{S}} y_{\mathbf{S}} f(\mathbf{S})$$

LP Formulation

$x_{e,i} = 1$ if item $e \in V$ is assigned to part $i \in [k]$

$y_{\mathbf{S}} = 1$ if $\mathbf{S} = (S_1, \ldots, S_k)$ is the k-tuple we select

Primal LP

$$\max \sum_{\mathbf{S}} y_{\mathbf{S}} f(\mathbf{S})$$

$$\forall e, i: \quad \sum_{\mathbf{S}:\, e \in S_i} y_{\mathbf{S}} = x_{e,i}$$

$$\sum_{\mathbf{S}} y_{\mathbf{S}} = 1$$

LP Formulation

$x_{e,i} = 1$ if item $e \in V$ is assigned to part $i \in [k]$

$y_{\mathbf{S}} = 1$ if $\mathbf{S} = (S_1, \ldots, S_k)$ is the k-tuple we select

Primal LP

$$\max \sum_{\mathbf{S}} y_{\mathbf{S}} f(\mathbf{S})$$

$$\forall e, i: \quad \sum_{\mathbf{S}:\, e \in S_i} y_{\mathbf{S}} = x_{e,i}$$

$$\sum_{\mathbf{S}} y_{\mathbf{S}} = 1$$

$$\forall e: \quad \sum_{i=1}^{k} x_{e,i} \leq 1$$

LP Formulation

$x_{e,i} = 1$ if item $e \in V$ is assigned to part $i \in [k]$

$y_{\mathbf{S}} = 1$ if $\mathbf{S} = (S_1, \ldots, S_k)$ is the k-tuple we select

Primal LP

$$\max \sum_{\mathbf{S}} y_{\mathbf{S}} f(\mathbf{S})$$

$$\forall e, i: \quad \sum_{\mathbf{S}:\, e \in S_i} y_{\mathbf{S}} = x_{e,i}$$

$$\sum_{\mathbf{S}} y_{\mathbf{S}} = 1$$

$$\forall e: \quad \sum_{i=1}^{k} x_{e,i} \le 1$$

$$\forall i: \quad \sum_{e \in V} x_{e,i} \le B_i$$

LP Formulation

$x_{e,i} = 1$ if item $e \in V$ is assigned to part $i \in [k]$

$y_{\mathbf{S}} = 1$ if $\mathbf{S} = (S_1, \ldots, S_k)$ is the k-tuple we select

Primal LP

$$\max \sum_{\mathbf{S}} y_{\mathbf{S}} f(\mathbf{S})$$

$$\forall e, i: \quad \sum_{\mathbf{S}:\, e \in S_i} y_{\mathbf{S}} = x_{e,i} \quad \leftarrow \alpha_{e,i}$$

$$\sum_{\mathbf{S}} y_{\mathbf{S}} = 1 \quad \leftarrow \beta$$

$$\forall e: \quad \sum_{i=1}^{k} x_{e,i} \le 1 \quad \leftarrow \gamma_e$$

$$\forall i: \quad \sum_{e \in V} x_{e,i} \le B_i \quad \leftarrow \phi_i$$

Dual LP

$$\min \; \beta + \sum_{e} \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \quad \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \ge f(\mathbf{S})$$

$$\forall e, i: \; \gamma_e + \phi_i \ge \alpha_{e,i}$$

Our Algorithm

Parameters: C, D

C, D : constants

Dual LP

$$\min \ \beta + \sum_{e} \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \ \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \geq f(\mathbf{S})$$

$$\forall e, i: \ \gamma_e + \phi_i \geq \alpha_{e,i}$$

Our Algorithm

Parameters: C, D

$\mathbf{S} = (S_1, \ldots, S_k) \leftarrow (\emptyset, \ldots, \emptyset)$

$\phi_i \leftarrow 0 \;\; \forall i \in [k]$

ϕ_i : threshold for marginal gains

Dual LP

$$\min \; \beta + \sum_e \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \; \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \geq f(\mathbf{S})$$

$$\forall e, i: \; \gamma_e + \phi_i \geq \alpha_{e,i}$$

Our Algorithm

Parameters: C, D

$\mathbf{S} = (S_1, \ldots, S_k) \leftarrow (\emptyset, \ldots, \emptyset)$

$\phi_i \leftarrow 0 \ \ \forall i \in [k]$

for each arriving item e:

Dual LP

$$\min \ \beta + \sum_e \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \ \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \geq f(\mathbf{S})$$

$$\forall e, i: \ \gamma_e + \phi_i \geq \alpha_{e,i}$$

Our Algorithm

Parameters: C, D

$\mathbf{S} = (S_1, \ldots, S_k) \leftarrow (\emptyset, \ldots, \emptyset)$

$\phi_i \leftarrow 0 \ \ \forall i \in [k]$

for each arriving item e:

$$i \leftarrow \arg\max_{j \in [k]} \left\{ \Delta_{e,j} f(\mathbf{S}) - \phi_j \right\}$$

discounted gain

Dual LP

$$\min \ \beta + \sum_e \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \ \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \geq f(\mathbf{S})$$

$$\forall e, i: \gamma_e + \phi_i \geq \alpha_{e,i}$$

Our Algorithm

Parameters: C, D

$\mathbf{S} = (S_1, \ldots, S_k) \leftarrow (\emptyset, \ldots, \emptyset)$

$\phi_i \leftarrow 0 \ \ \forall i \in [k]$

for each arriving item e:

$$i \leftarrow \arg\max_{j \in [k]} \left\{ \Delta_{e,j} f(\mathbf{S}) - \phi_j \right\}$$

if $\Delta_{e,i} f(\mathbf{S}) - \phi_i \geq 0$:

$S_i \leftarrow S_i \cup \{e\}$

remove earliest item to ensure $|S_i| \leq B_i$

Dual LP

$$\min \ \beta + \sum_e \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \ \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \geq f(\mathbf{S})$$

$$\forall e, i: \ \gamma_e + \phi_i \geq \alpha_{e,i}$$

Our Algorithm

Parameters: C, D

$\mathbf{S} = (S_1, \ldots, S_k) \leftarrow (\varnothing, \ldots, \varnothing)$

$\phi_i \leftarrow 0 \quad \forall i \in [k]$

for each arriving item e:

$$i \leftarrow \arg\max_{j \in [k]} \left\{ \Delta_{e,j} f(\mathbf{S}) - \phi_j \right\}$$

if $\Delta_{e,i} f(\mathbf{S}) - \phi_i \geq 0$:

$$\gamma_e \leftarrow \Delta_{e,i} f(\mathbf{S}) - \phi_i$$

$$S_i \leftarrow S_i \cup \{e\}$$

remove earliest item to ensure $|S_i| \leq B_i$

Dual LP

$$\min \ \beta + \sum_e \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \ \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \geq f(\mathbf{S})$$

$$\forall e, i: \ \gamma_e + \phi_i \geq \alpha_{e,i}$$

Our Algorithm

Parameters: C, D

$\mathbf{S} = (S_1, \ldots, S_k) \leftarrow (\varnothing, \ldots, \varnothing)$

$\phi_i \leftarrow 0 \quad \forall i \in [k]$

for each arriving item e:

$$i \leftarrow \arg\max_{j \in [k]} \left\{ \Delta_{e,j} f(\mathbf{S}) - \phi_j \right\}$$

if $\Delta_{e,i} f(\mathbf{S}) - \phi_i \geq 0$:

$$\gamma_e \leftarrow \Delta_{e,i} f(\mathbf{S}) - \phi_i$$

$$\phi_i \leftarrow \left(1 + \frac{D}{B_i}\right) \phi_i + \frac{C}{B_i} \gamma_e$$

$$S_i \leftarrow S_i \cup \{e\}$$

remove earliest item to ensure $|S_i| \leq B_i$

return $\mathbf{S}$

Dual LP

$$\min \; \beta + \sum_e \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \; \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \geq f(\mathbf{S})$$

$$\forall e, i: \; \gamma_e + \phi_i \geq \alpha_{e,i}$$

Our Algorithm

Parameters: C, D

$\mathbf{S} = (S_1, \ldots, S_k) \leftarrow (\emptyset, \ldots, \emptyset)$

$\phi_i \leftarrow 0 \ \ \forall i \in [k]$

for each arriving item e:

$$i \leftarrow \arg\max_{j \in [k]} \left\{ \Delta_{e,j} f(\mathbf{S}) - \phi_j \right\}$$

if $\Delta_{e,i} f(\mathbf{S}) - \phi_i \geq 0$:

$$\gamma_e \leftarrow \Delta_{e,i} f(\mathbf{S}) - \phi_i$$

$$\phi_i \leftarrow \left(1 + \frac{D}{B_i}\right) \phi_i + \frac{C}{B_i} \gamma_e$$

$$S_i \leftarrow S_i \cup \{e\}$$

remove earliest item to ensure $|S_i| \leq B_i$

return $\mathbf{S}$

Dual LP

$$\min \ \beta + \sum_e \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \ \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \geq f(\mathbf{S})$$

$$\forall e, i: \ \gamma_e + \phi_i \geq \alpha_{e,i}$$

$\alpha_{e,i} = \gamma_e + \phi_i$

$\mathbf{X} =$ all items added to $\mathbf{S}$

$\beta = f(\mathbf{X})$

Our Algorithm

Parameters: C, D

$\mathbf{S} = (S_1, \ldots, S_k) \leftarrow (\emptyset, \ldots, \emptyset)$

$\phi_i \leftarrow 0 \ \ \forall i \in [k]$

for each arriving item e:

$$i \leftarrow \arg\max_{j \in [k]} \left\{ \Delta_{e,j} f(\mathbf{S}) - \phi_j \right\}$$

if $\Delta_{e,i} f(\mathbf{S}) - \phi_i \geq 0$:

$$\gamma_e \leftarrow \Delta_{e,i} f(\mathbf{S}) - \phi_i$$

$$\phi_i \leftarrow \left(1 + \frac{D}{B_i}\right) \phi_i + \frac{C}{B_i} \gamma_e$$

$$S_i \leftarrow S_i \cup \{e\}$$

remove earliest item to ensure $|S_i| \leq B_i$

return $\mathbf{S}$

Dual LP

$$\min \ \beta + \sum_e \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{S}: \ \sum_{i=1}^{k} \sum_{e \in S_i} \alpha_{e,i} + \beta \geq f(\mathbf{S})$$

Ensure dual sol is feasible

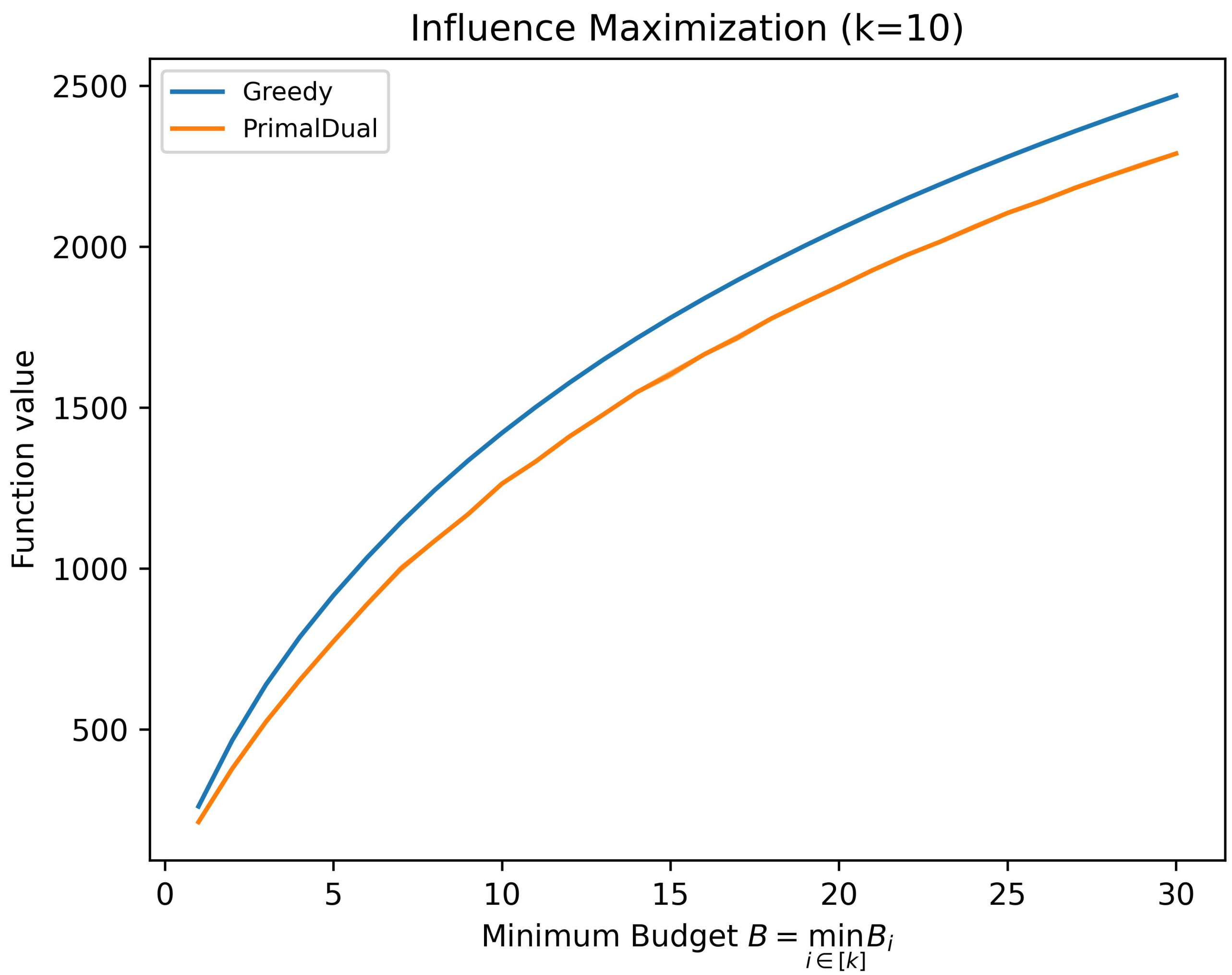
$\alpha_{e,i} = \gamma_e + \phi_i$

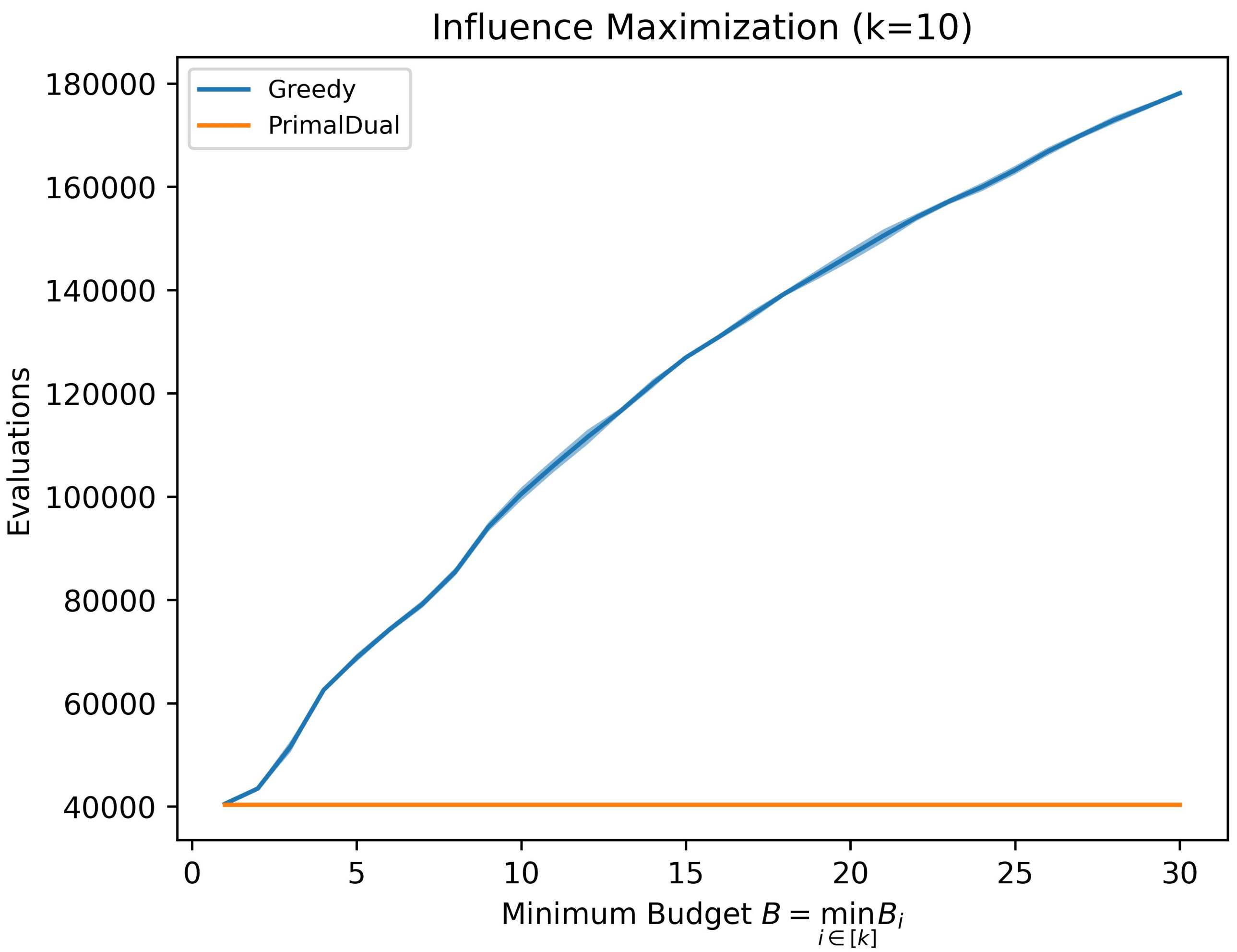
$\mathbf{X} =$ all items added to $\mathbf{S}$

$\beta = f(\mathbf{X})$

Experimental Results

Synthetic instances of influence maximization

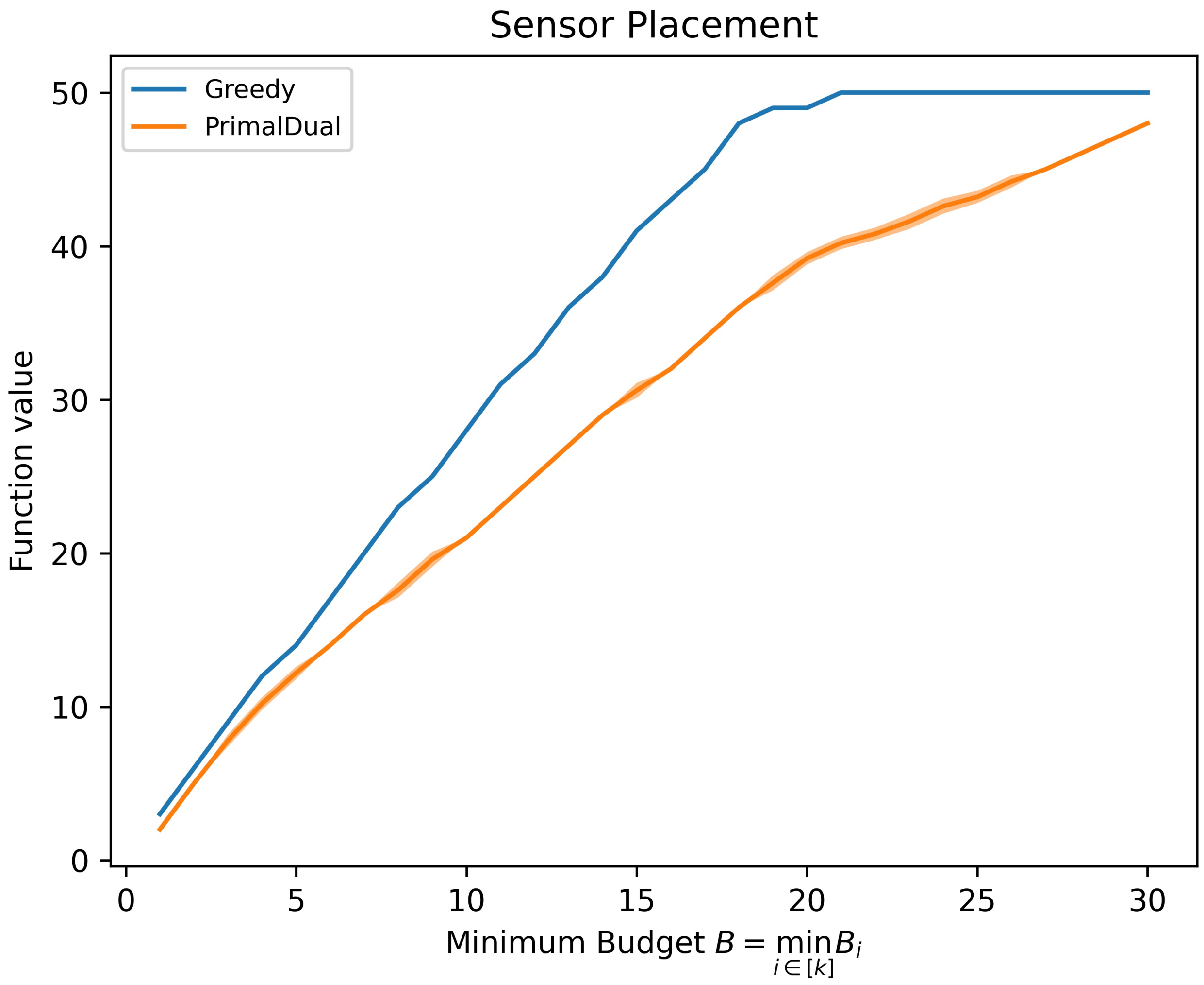


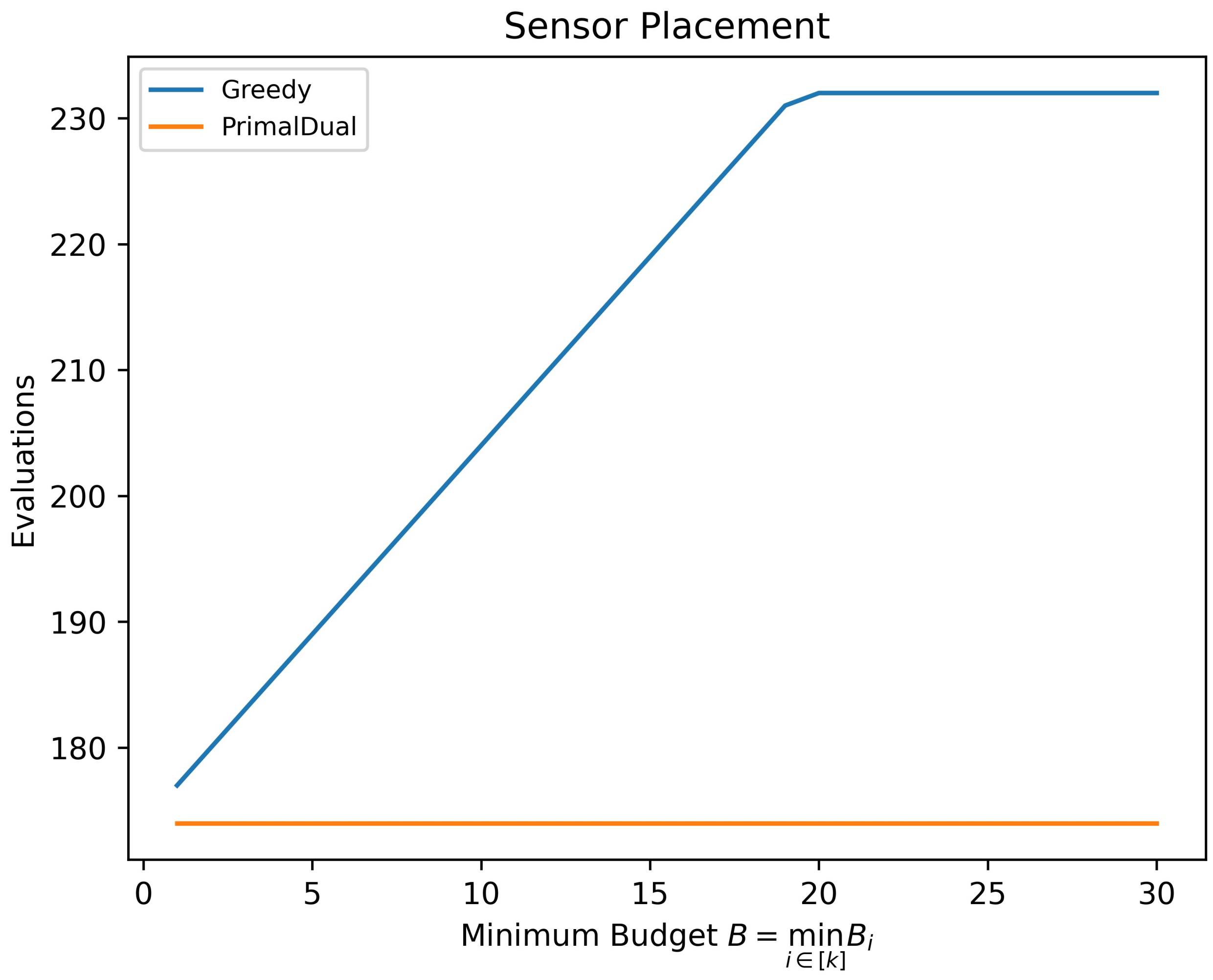


$k = 10$ different topics

Experimental Results

Sensor placement on the Intel Lab dataset [Bodik et al. '04]





$k = 3$ types of sensors