Cost Ratio Aware Algorithm for Representative Subset Selection

Tong Cheng
Nanyang Technological University
Singapore
tong030@e.ntu.edu.sg

Xueyan Tang
Nanyang Technological University
Singapore
asxytang@ntu.edu.sg

Abstract

Representative subset selection is an important problem in web data mining and machine learning, which is commonly modeled as monotone submodular maximization under a knapsack constraint with a cost function c and a budget b. Despite extensive studies, it remains unclear whether there exists any deterministic combinatorial algorithm that can achieve an approximation ratio $\alpha > 1/2$ with subquadratic query complexity. In this paper, we introduce the max/min cost ratio of all elements $r := \max_e c(e)/\min_e c(e)$ as a structural parameter that facilitates the exploration of this gap with a hardness result and several intuitive insights for algorithm design. We first show that when $\lfloor b/r \rfloor = 1$, no deterministic combinatorial algorithm with $o(n^2)$ query complexity can achieve an approximation ratio $\alpha > 1/2$. When $\lfloor b/r \rfloor \geq 2$, we identify non-trivial regions in the (r, b)-plane where thresholding greedy-based algorithms achieve $(5/9 - \varepsilon)$-approximation ratios with $O((n/\varepsilon)\log(n/\varepsilon))$ query complexities. Finally, we propose a Cost Ratio Aware (CRA) algorithm that switches between these algorithms based on the property of (r, b). Experiments on four web-based applications show that CRA outperforms other baselines within the identified regions, implying that the cost ratio is an effective indicator of algorithmic performance.

CCS Concepts

• **Information systems** → **Web mining**; • **Mathematics of computing** → **Submodular optimization and polymatroids**.

Keywords

web data mining; machine learning; submodular maximization; knapsack constraint; approximation ratio

ACM Reference Format:
Tong Cheng and Xueyan Tang. 2026. Cost Ratio Aware Algorithm for Representative Subset Selection. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3774904.3792869

1 Introduction

Representative subset selection is a core application in modern web systems with well-known applications such as data summarization, sensor placement, influence maximization and resource allocation.

These applications could be formulated as the monotone submodular maximization under a knapsack constraint (SMK) problem. Let V be the ground set with size n. The goal of SMK is to choose a subset $S \subseteq V$ with $c(S) \leq b$ that maximizes a utility function $f(S)$ where $f : 2^V \to \mathbb{R}_{\geq 0}$ is a non-negative monotone submodular function, $c : V \to \mathbb{R}_{>0}$ is a modular cost function and $b > 0$ is the budget. This formulation captures the ubiquitous trade-off between coverage (or influence) and resource limitations in web-based systems. For instance, the budget b can represent CPU limits for scoring and re-ranking, bandwidth or memory limits for serving content. In the value oracle model, an algorithm makes a query by inputting a set S, and the oracle returns the value $f(S)$. The query complexity of the algorithm is the total number of sets S for which the algorithm requests the value $f(S)$ from the oracle.

In the offline setting, Sviridenko [5] proposes an algorithm achieving a $(1 - e^{-1})$-approximation ratio for SMK by enumerating all three-element subsets with $O(n^5)$ query complexity. Feldman et al. [3] improve this enumeration idea by showing that enumerating two-element subsets also achieves $(1 - e^{-1})$-approximation with $O(n^4)$ query complexity. They further present a One-Set Enumeration algorithm achieving approximately $0.97(1-e^{-1})$-approximation with $O(n^3)$ queries. All these algorithms could be accelerated from $O(n^m)$ $(m \in \{3, 4, 5\})$ to $\tilde{O}(n^{m-1}/\varepsilon)$ using the thresholding technique in [1] at the cost of worsening the approximation ratio by a factor of $(1 - \varepsilon)$ where the $\tilde{O}$ notation disregards poly-logarithmic terms. Despite offering strong approximation guarantees, these algorithms suffer from a quadratic or higher query complexity, rendering them impractical for web-based applications. In contrast, algorithms with subquadratic query complexities are favorable in web-based applications.

For more efficient algorithms, Ene et al. [2] propose a multilinear extension framework that achieves a $(1 - e^{-1} - \varepsilon)$-approximation ratio in near-linear time. However, since the query complexity becomes extremely large by ε and the cost of evaluating multilinear extension is expensive, their method is impractical in real-world applications. A complementary line of work focuses on more practical algorithms without leveraging continuous extensions (we call this algorithm as *combinatorial*). The Greedy+Max algorithm, proposed by Avdiukhin et al. [6], achieves a $(1/2)$-approximation ratio with $O(n^2)$ query complexity and can be accelerated to $\tilde{O}(n/\varepsilon)$. Subsequently, Li et al. [4] propose Fast Threshold Greedy with Post-Processing (FTG+PP), a near-linear-time algorithm that guarantees a $(1/2 - \varepsilon)$-approximation ratio for SMK. However, these two near-linear-time algorithms have lower approximation ratios than the previous works [2, 3, 5].

These observations raise a fundamental algorithmic question: *Does there exist any deterministic combinatorial algorithm that could achieve an approximation ratio $\alpha > 1/2$ for SMK with subquadratic query complexity?* To facilitate the investigation of this question, we introduce a simple structural parameter called the cost ratio and study how it affects both hardness and algorithmic performance.

WWW '26, Dubai, United Arab Emirates

ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792869

Specifically, given a cost function c, we define the cost ratio as $r := \frac{\max_{e \in V} c(e)}{\min_{e \in V} c(e)}$ (max/min cost ratio among all elements). Without loss of generality, we can normalize a knapsack constraint (c, b) such that $\min_{e \in V} c(e) = 1$ and $r = \max_{e \in V} c(e)$.

Contributions. Our first contribution is a hardness result. When $\lfloor b/r \rfloor = 1$, no deterministic combinatorial algorithm with $o(n^2)$ query complexity can achieve an approximation ratio strictly larger than $1/2$. This result demonstrates the near-optimality of the $1/2 - \varepsilon$ approximation ratios achieved by existing near-linear-time combinatorial algorithms, such as Greedy+Max (thresholded version) and FTG+PP.

We then move beyond this worst-case barrier and focus on instances where $\lfloor b/r \rfloor \geq 2$. On the algorithmic side, we analyze a *Greedy Power Density* algorithm which iteratively adds feasible elements maximizing $f(e \mid S)/c(e)^\theta$, given a power parameter $\theta \in [0, 1]$. We derive an approximation ratio of this algorithm as a function of (r, b, θ) and prove that the best approximation ratio is achieved at the endpoints $\theta \in \{0, 1\}$. By analyzing the endpoint cases, we identify three regions (A, B, C) in the (r, b)-plane where Greedy Power Density achieves a $(5/9)$-approximation ratio. These three regions cover at least 37.8% area of the whole region where $\lfloor b/r \rfloor \geq 2$.

Building on these insights, we propose a Cost Ratio Aware (CRA) algorithm that switches between two thresholded power-density rules (with $\theta = 0$ and $\theta = 1$) and a near-linear-time baseline (FTG+PP), based on the observable pair (r, b). When $(r, b) \in A \cup B \cup C$, CRA achieves a $(5/9 - \varepsilon)$ approximation ratio with $O((n/\varepsilon)\log(n/\varepsilon))$ query complexity. Otherwise, it achieves a $(1/2 - \varepsilon)$ approximation ratio with $O((n/\varepsilon)\log(1/\varepsilon))$ query complexity.

Finally, we evaluate CRA on four web-based applications and observe that CRA consistently outperforms near-linear-time baselines when $(r, b) \in A \cup B \cup C$ while remaining competitive elsewhere, which is consistent with our theoretical results.

2 Theoretical Results

By definition, a set function f is monotone if for all $A \subseteq B \subseteq V$, it holds that $f(A) \leq f(B)$, and is submodular if for all $A \subseteq B \subseteq V$ and any element $e \in V \setminus B$, it holds that $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$. The marginal gain of adding an element e to a set S is defined as $f(e \mid S) := f(S \cup \{e\}) - f(S)$ for any $S \subseteq V$ and $e \in V \setminus S$. Let $OPT := \arg\max_{S \subseteq V : c(S) \leq b} f(S)$.

In the following, we categorize SMK problem instances by the cost ratio and budget pair (r, b) to analyze their complexity and develop efficient algorithms.

2.1 Hardness Result

Theorem 2.1. *When $\lfloor b/r \rfloor = 1$, no deterministic combinatorial algorithm with $o(n^2)$ query complexity could achieve an approximation ratio of $\alpha > 1/2$.*

Proof. Fix an integer $n \geq 2$ and let the ground set be a disjoint union $V = H \cup L$ with $|H| = \lfloor n/2 \rfloor \geq 1$ and $|L| = n - |H| \geq 1$, where H is the set of heavy elements and L is the set of light elements. We set a budget b with $2 < b < 3$ and define a normalized cost function such that $c(\ell) = 1$ for all $\ell \in L$ and $c(w) \in (b/2, b-1]$ for all $w \in H$. Since $\min_{e \in V} c(e) = 1$ and $r = \max_{e \in V} c(e) \in (b/2, b-1]$, it follows that $\lfloor b/r \rfloor = 1$.

With the above settings, for any $e \in V$, it holds that $c(e) \leq b$. For any $\{e_1, e_2\} \subseteq V$, if $c(\{e_1, e_2\}) > b$, it holds that $\{e_1, e_2\} \subseteq H$. It also holds that $c(S) > b$ for any subset $S \subseteq V$ with $|S| \geq 3$. Based on these three observations, we conclude that the feasible solution set is composed of all singleton elements, light-light pairs, and heavy-light pairs.

Let $\mathcal{A}$ be any deterministic combinatorial algorithm and define $Q := \{(w, \ell) \in H \times L : \mathcal{A} \text{ queries } \{w, \ell\}\}$. Since the query complexity of $\mathcal{A}$ is $o(n^2)$ while $|H \times L| = \Theta(n^2)$, it follows that there exists a pair $(w^\star, \ell^\star) \in (H \times L) \setminus Q$. Using two elements $w^\star \in H$ and $\ell^\star \in L$, define $f : 2^V \to \mathbb{R}_{\geq 0}$ as $f(S) = 0$ if $S = \emptyset$; $f(S) = 2$ if $\{w^\star, \ell^\star\} \subseteq S$; and $f(S) = 1$ otherwise. Given objective f, cost function c and budget b, the optimal solution is $OPT = \{w^\star, \ell^\star\}$ with $f(OPT) = 2$.

Let $S_{\mathcal{A}}$ denote the output of algorithm $\mathcal{A}$. By the definition of Q, no query made by $\mathcal{A}$ ever contains both $w^\star$ and $\ell^\star$. Hence, $f(S_{\mathcal{A}}) \leq 1$, implying that the approximation ratio of $\mathcal{A}$ is at most $1/2$. Therefore, any deterministic combinatorial algorithm that achieves an approximation ratio $\alpha > 1/2$ with $\lfloor b/r \rfloor = 1$ must have $\Omega(n^2)$ oracle calls in the worst case. □

Theorem 2.1 shows that it is impossible to find a deterministic combinatorial algorithm with subquadratic query complexity that could achieve approximation ratio strictly larger than $1/2$ when $\lfloor b/r \rfloor = 1$. Hence, in the following we focus on the SMK problem instance satisfying $\lfloor b/r \rfloor \geq 2$.

2.2 Greedy Power Density

Given a parameter $\theta \in [0, 1]$, *Greedy Power Density* (Algorithm 1) starts from $S = \emptyset$ and repeatedly adds a feasible element maximizing the power density $f(e \mid S)/c(e)^\theta$. It stops when no element could be added into the current solution set due to budget constraint.

Algorithm 1 Greedy Power Density (GPD)

Input: θ

```
1: S ← ∅; Ṽ ← V;
2: while Ṽ ≠ ∅ do
3:     u ← arg max_{e∈Ṽ\S} f(e | S)/c(e)^θ;
4:     if c(S) + c(u) ≤ b then S ← S ∪ {u}
5:     Ṽ ← Ṽ \ {u};
6: return S
```

Let $A_1 := \lfloor b \rfloor r^\theta$, $A_2 := \lfloor b \rfloor^{1-\theta} b^\theta$, $B_1 := \lfloor b/r \rfloor$ and $B_2 := \frac{\alpha_1(r,b)}{r^{1-\theta}}$ where $\alpha_1(r, b) = b - r$ if $b < r\lfloor b \rfloor$; otherwise $\alpha_1(r, b) = \lfloor b \rfloor$.

Theorem 2.2. *For any $\theta \in [0, 1]$, Greedy Power Density achieves an approximation ratio of $1 - e^{-\gamma(\theta)}$ where $\gamma(\theta) := \frac{\max\{B_1, B_2\}}{\min\{A_1, A_2\}}$. In particular, it achieves $1 - \left(1 - \frac{1}{\min\{A_1, A_2\}}\right)^{\max\{B_1, B_2\}}$-approximation if $\theta = 0$.*

Proof Sketch. Our analysis tracks the iterative form of the optimality gap $f(OPT) - f(S)$ along the greedy trajectory by comparing each step's gain to an upper bound on the residual "power cost" of OPT. Specifically, for the intermediate solution sets $S_1, \ldots, S_l$ of GPD, our analysis compares each marginal gain $f(S_i) - f(S_{i-1})$

to the remaining "power cost" of OPT. Let $C_i := \sum_{e \in OPT \setminus S_{i-1}} c(e)^\theta$ and $D := \sum_{i=1}^{l'} c(u_i)^\theta$ where $u_i := S_i \setminus S_{i-1}$ and l' is the largest index such that, up to the point when $S_{l'}$ is constructed, no element from OPT has been selected by maximizing power density but not added to the solution set due to the budget constraint. By monotonicity and submodularity, GPD's choice of u_i gives $f(S_i) - f(S_{i-1}) \geq \frac{c(u_i)^\theta}{C_i}(f(OPT) - f(S_{i-1}))$, implying that the gap $f(OPT) - f(S_i)$ contracts multiplicatively at a rate controlled by $c(u_i)^\theta / C_i$. Bounding the denominator C_i and the total "power cost" D by the definition of cost ratio and budget yields $C_i \leq \min\{A_1(\theta), A_2(\theta)\}$ and $D \geq \max\{B_1, B_2(\theta)\}$. Substituting these envelopes into the above recursion proves Theorem 2.2.[1] □

Lemma 2.3. *$\gamma(\theta)$ achieves its maximum when $\theta \in \{0, 1\}$.*

Proof. Let $s := \min\{r, b/\lfloor b \rfloor\} \geq 1$ and $c_0 := B_1$, $c_1 := \alpha_1(r, b)/r$. It holds that $\min\{A_1(\theta), A_2(\theta)\} = \lfloor b \rfloor s^\theta$ and $B_2(\theta) = c_1 r^\theta$. (Case 1): if $c_0 \geq c_1 r^\theta$, we have $\gamma(\theta) = \frac{c_0}{\lfloor b \rfloor} s^{-\theta}$. Since $s \geq 1$, $\gamma(\theta)$ is non-increasing with θ. (Case 2): if $c_0 \leq c_1 r^\theta$, we have $\gamma(\theta) = \frac{c_1}{\lfloor b \rfloor}\left(\frac{r}{s}\right)^\theta$. Since $r/s \geq 1$, $\gamma(\theta)$ is non-decreasing with θ. Hence, $\gamma(\theta)$ achieves its maximum at the endpoints for every (r, b) pair. □

Define three regions in the (r, b)-plane as $A := \{(r, b) : r \leq b/\lfloor b \rfloor\}$, $B := \{(r, b) : b/r \geq (1 - \ln(9/4))^{-1}\}$ and $C := \{(r, b) : \lfloor b/r \rfloor \in \{2, 3, 4, 5\}, \lfloor b \rfloor \leq \lfloor b/r \rfloor + 1\}$.

Lemma 2.4. *If $(r, b) \in A \cup C$, then GPD with $\theta = 0$ achieves a $(5/9)$-approximation ratio. If $(r, b) \in B$, then GPD with $\theta = 1$ achieves a $(5/9)$-approximation ratio.*

Proof Sketch. This lemma is derived by setting $\theta = 0$ and $\theta = 1$ on the approximation ratio of GPD from Theorem 2.2 and simplifying the resulting inequalities.[1] □

We quantify the area ratio β of the region $A \cup B \cup C$. For any value $M > 1$, define $S_M := \{(r, b) : 1 \leq r \leq b \leq M, \lfloor b/r \rfloor \geq 2\}$ and $R_M := (A \cup B \cup C) \cap S_M$. Let $|\cdot|$ denote the two-dimensional Lebesgue measure on the (r, b)-plane. Then β is defined as $\beta := \lim_{M\to\infty}(|R_M|/|S_M|)$.

Lemma 2.5. *It holds that $\beta \geq 2(1 - \ln(9/4)) \approx 0.378$.*

Proof. For simplicity, let $\lambda := (1 - \ln(9/4))^{-1}$. We first compute the area of S_M. The constraint $\lfloor b/r \rfloor \geq 2$ is equivalent to $b \geq 2r$, so on S_M we have $1 \leq r \leq M/2$ and $2r \leq b \leq M$. Hence, $|S_M| = \int_{r=1}^{M/2} \int_{b=2r}^{M} db dr = \frac{M^2}{4} + o(M^2)$.

Next, we lower bound $|R_M|$ by $|B \cap S_M|$. For $M \geq \lambda$, it holds that $B \cap S_M = \{(r, b) : \lambda \leq b \leq M, 1 \leq r \leq b/\lambda\}$. Hence, $|B \cap S_M| = \int_{b=\lambda}^{M} \int_{r=1}^{b/\lambda} dr db = \frac{M^2}{2\lambda} + o(M^2)$. Since $|R_M| \geq |B \cap S_M|$, it follows that $\frac{|R_M|}{|S_M|} \geq \frac{|B \cap S_M|}{|S_M|} = \frac{\frac{M^2}{2\lambda} + o(M^2)}{\frac{M^2}{4} + o(M^2)}$ which implies that $\lim_{M\to\infty} \frac{|R_M|}{|S_M|} \geq \frac{1/(2\lambda)}{1/4} = \frac{2}{\lambda} \approx 0.378$. □

Lemma 2.5 suggests that GPD achieves a $(5/9)$-approximation ratio for at least 37.8% SMK problem instances of all (r, b) pairs with $\lfloor b/r \rfloor \geq 2$.

Algorithm 2 Threshold Greedy Power Density (TGPD)

Input: θ, ε

1: $S \leftarrow \emptyset$; $\tau_0 \leftarrow \max_{e \in V}(f(\{e\})/c(e)^\theta)$; $\tau \leftarrow \tau_0$;
2: **while** $\tau \geq \varepsilon\tau_0/n$ **do**
3:     **for** each $e \in V$ **do**
4:         **if** $c(S) + c(e) \leq b$ and $f(e \mid S)/c(e)^\theta \geq \tau$ **then**
5:             $S \leftarrow S \cup \{e\}$;
6:     $\tau \leftarrow (1 - \varepsilon) \cdot \tau$;
7: **return** S

2.3 Cost Ratio Aware Algorithm

According to Algorithm 1, GPD has $O(n^2)$ query complexity. To accelerate GPD into subquadratic query complexity, we apply the thresholding technique from [1] on GPD to obtain Threshold Greedy Power Density (Algorithm 2).

Lemma 2.6. *TGPD achieves an approximation ratio of $(1-\varepsilon)\alpha$ with $O((n/\varepsilon)\log(n/\varepsilon))$ query complexity where α is the approximation ratio of GPD.*

Algorithm 3 Cost Ratio Aware (CRA)

Input: ε

1: normalize (c, b) and compute cost ratio r
2: **if** $(r, b) \in A \cup C$ **then return** TGPD$(0, \varepsilon)$
3: **else if** $(r, b) \in B$ **then return** TGPD$(1, \varepsilon)$
4: **else return** FTG+PP(ε) [4]

Combining Lemma 2.4, Lemma 2.6 and the approximation ratio of FTG+PP from [4], we propose a Cost Ratio Aware algorithm (Algorithm 3) and obtain the following theorem:

Theorem 2.7. *For SMK problem, CRA has $O((n/\varepsilon)\log(n/\varepsilon))$ query complexity and its output S_{CRA} satisfies:*

(1) *If $(r, b) \in A \cup B \cup C$, $f(S_{\text{CRA}}) \geq (5/9 - \varepsilon)f(OPT)$.*
(2) *Otherwise, $f(S_{\text{CRA}}) \geq (1/2 - \varepsilon)f(OPT)$.*

3 Experiments

In this section, we evaluate the performance of CRA on four web-based applications and compare CRA with three baselines across a range of cost ratios and budgets. Our experiments are designed to test two hypotheses: (i) for SMK problem instances with $(r, b) \in A \cup B \cup C$, CRA outperforms baselines whose approximation ratios are lower than CRA; (ii) otherwise, if $(r, b) \notin A \cup B \cup C$, CRA's performance does not dominate baselines with the same approximation ratio and hence CRA loses its advantage.

We conduct experiments on four web-based applications: movie recommendation, image summarization, facility location and revenue maximization, following the same settings as previous work [4]. Each application specifies an objective function f and a cost function c. We sample the cost ratio and budget pair $(r^\star, b^\star)$ uniformly from $\{(r, b) : 1 \leq r \leq b \leq 300\}$. For each application with a cost function c, we transform the cost $c(e)$ to $\tilde{c}(e) = 1 + (r^\star - 1) \cdot \frac{c(e) - c_{\min}}{c_{\max} - c_{\min}}$ and then set the budget to $b^\star$. We classify problem instances into

[1] Full proofs can be found at: https://github.com/tongch8819/CostRatioAwareSMK.

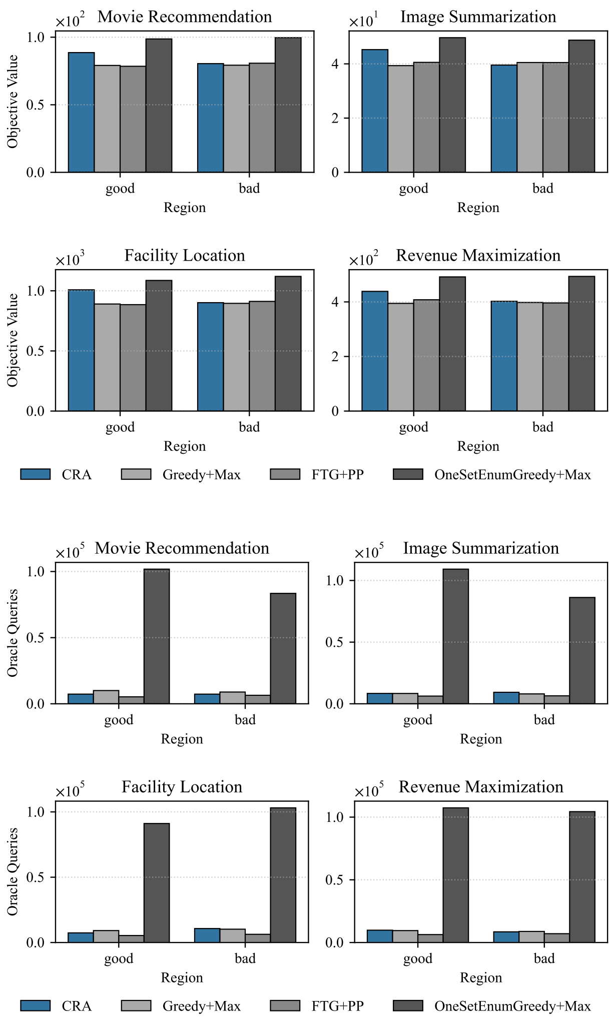

Figure 1: Average objective value and number of oracle queries across two regions and four applications.

two categories: (i) good when $(r^\star, b^\star) \in A \cup B \cup C$; (ii) bad otherwise. Our experiments are conducted on a 36-core Linux server with an Intel Core i9-10980XE CPU @ 3.00GHz and 125GB of RAM.

We compare CRA against three baselines: 1) *Greedy+Max* proposed by [6]; 2) *FTG+PP* proposed by [4]; 3) *OneSetEnumGreedy+Max* proposed by [3]. We implement Greedy+Max and OneSetEnumGreedy+Max using the thresholding technique from [1] to ensure comparison fairness. The accuracy parameter ε of all algorithms are set to 10^{-3}. For each application and sampled pair $(r^\star, b^\star)$, we run CRA and other baselines for the problem instance, and record the objective value of the final output and the number of oracle queries. For each region type (i.e., good or bad), we compute the average objective value and number of oracle queries, which are illustrated in Figure 1.

We observe that in both regions OneSetEnumGreedy+Max achieves the highest average objective values while its number of oracle queries are way much higher than other methods. In the good region, CRA achieves higher average objective values than Greedy+Max and FTG+PP while having almost the same average number of oracle calls as Greedy+Max and FTG+PP. In the bad region, CRA achieves almost the same average objective values as Greedy+Max and FTG+PP and loses its advantage. These observations are consistent with our theoretical results, suggesting that if the SMK problem instance lies in the good region, we could use CRA to achieve better performance without increasing the overhead of oracle queries.

4 Conclusion

In this paper, we have studied the problem of monotone submodular maximization under a knapsack constraint and aim to find a deterministic combinatorial algorithm with subquadratic query complexity and approximation ratio strictly larger than $1/2$. In order to investigate this problem, we introduce a structural parameter called cost ratio. With this tool, we firstly prove that when $\lfloor b/r \rfloor = 1$, no deterministic combinatorial algorithm with subquadratic query complexity can achieve an approximation ratio $\alpha > 1/2$. Furthermore, we propose a Greedy Power Density algorithm and through dedicated analysis identify three regions A, B, C in the (r, b)-plane where Greedy Power Density achieves a $(5/9)$-approximation ratio. These regions cover at least 37.8% of all (r, b) pairs with $\lfloor b/r \rfloor \geq 2$. Building on this theoretical result, we design a Cost Ratio Aware algorithm that switches between different sub-procedures based on the property of (r, b) with near-linear query complexity, achieving $(5/9 - \varepsilon)$-approximation in $A \cup B \cup C$ and $(1/2 - \varepsilon)$-approximation elsewhere. Experiments on four web-based applications shows that CRA outperforms baselines with a lower approximation ratio and near-linear query complexity in the favorable region and remains competitive outside the favorable region.

Acknowledgments

This research is supported by the Ministry of Education, Singapore, under its AcRF Tier 2 (Award MOE-T2EP20122-0007).

References

[1] Ashwinkumar Badanidiyuru and Jan Vondrák. 2014. Fast algorithms for maximizing submodular functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms* (Portland, Oregon) *(SODA '14)*. Society for Industrial and Applied Mathematics, USA, 1497–1514.

[2] Alina Ene and Huy L. Nguyen. 2019. A Nearly-Linear Time Algorithm for Submodular Maximization with a Knapsack Constraint. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi (Eds.), Vol. 132. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 53:1–53:12. doi:10.4230/LIPIcs.ICALP.2019.53

[3] Moran Feldman, Zeev Nutov, and Elad Shoham. 2022. Practical Budgeted Submodular Maximization. *Algorithmica* 85, 5 (Dec. 2022), 1332–1371. doi:10.1007/s00453-022-01071-2

[4] Wenxin Li, Moran Feldman, Ehsan Kazemi, and Amin Karbasi. 2022. Submodular maximization in clean linear time. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1270, 15 pages.

[5] Maxim Sviridenko. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters* 32, 1 (Jan. 2004), 41–43. doi:10.1016/S0167-6377(03)00062-2

[6] Grigory Yaroslavtsev, Samson Zhou, and Dmitrii Avdiukhin. 2020. "Bring Your Own Greedy"+Max: Near-Optimal 1/2-Approximations for Submodular Knapsack. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 108)*, Silvia Chiappa and Roberto Calandra (Eds.). PMLR, 3263–3274. https://proceedings.mlr.press/v108/yaroslavtsev20a.html