Definition 1

Definition 1 *(The Static Problem).* Given a monotone objective function $f : 2^V \to \mathbb{R}^+$, a monotone cost function $c : 2^V \to \mathbb{R}^+$ and a budget B, the goal is to compute a solution X such that

$$X = \arg\max_{Y \subseteq V} f(Y) \text{ s.t. } c(Y) \leq B.$$

Definition 2

Definition 2 *(The Dynamic Problem)*. Let X be a ϕ-approximation for the problem in Definition 1. The dynamic problem is given by a sequence of changes where in each change the current budget B changes to $B^* = B + d$, $d \in \mathbb{R}_{\geq -B}$. The goal is to compute a ϕ-approximation X' for each newly given budget B^*.

Definition 3

submodularity ratio :

$$\alpha_f = \min_{X \subseteq Y, v \notin Y} \frac{f(X \cup \{v\}) - f(X)}{f(Y \cup \{v\}) - f(Y)}.$$

curvature

$$\kappa_f = 1 - \min_{v \in V} \frac{f(V) - f(V \setminus \{v\})}{f(v)}$$

approximate cost function $\hat{c}$

$$\alpha_c = \alpha_{\hat{c}} \text{ and } c(X) \leq \hat{c}(X) \leq \psi c(X),$$

approximate optimal solution $\tilde{X}$

$$\max\{f(X) | c(X) \leq \alpha \frac{B(1+\alpha(K_c-1)(1-k_c))}{\psi(n) K_c}\}.$$

Algorithm 1

Algorithm 1: Generalized Greedy Algorithm.

input: Initial budget constraint B.

$X \leftarrow \emptyset$;

$V' \leftarrow V$;

repeat

  $v^* \leftarrow \arg\max_{v \in V'} \frac{f(X \cup \{v\}) - f(X)}{\hat{c}(X \cup \{v\}) - \hat{c}(X)}$;

  if $\hat{c}(X \cup \{v^*\}) \leq B$ **then**

    $X \leftarrow X \cup \{v^*\}$;

  $V' \leftarrow V' \setminus \{v^*\}$;

until $V' \leftarrow \emptyset$;

$v^* \leftarrow \arg\max_{v \in V; \hat{c}(v) \leq B} f(v)$;

return $\arg\max_{S \in \{X, v^*\}} f(S)$

Algorithm 2

Algorithm 2: Adaptive Generalized Greedy Algorithm.

Let B^* be the new budget;
if $B^* < B$ **then**
    while $\hat{c}(X) > B^*$ **do**
        $v^* \leftarrow \arg\min_{v \in X} \frac{f(X) - f(X \setminus \{v\})}{\hat{c}(X) - \hat{c}(X \setminus \{v\})}$;
        $X \leftarrow X \setminus \{v^*\}$;
else if $B^* > B$ **then**
    $V' \leftarrow V \setminus X$;
    repeat
        $v^* \leftarrow \arg\max_{v \in V'} \frac{f(X \cup \{v\}) - f(X)}{\hat{c}(X \cup \{v\}) - \hat{c}(X)}$;
        if $\hat{c}(X \cup \{v^*\}) \leq B^*$ **then**
            $X \leftarrow X \cup \{v^*\}$;
        $V' \leftarrow V' \setminus \{v^*\}$;
    until $V' \leftarrow \emptyset$;
$v^* \leftarrow \arg\max_{v \in V; \hat{c}(v) \leq B^*} f(v)$;
return $\arg\max_{S \in \{X, v^*\}} f(S)$

Algorithm 3

Algorithm 3: POMC Algorithm.

input: Initial budget constraint B, time T

$X \leftarrow \{0\}^n$;

Compute $(f_1(X), f_2(X))$;

$P \leftarrow \{x\}$;

$t \leftarrow 0$;

while $t < T$ **do**

    Select X from P uniformly at random;

    $X' \leftarrow$ flip each bit of X with probability $\frac{1}{n}$;

    Compute $(f_1(X'), f_2(X'))$;

    if $\nexists Z \in P$ *such that* $Z \succ X'$ **then**

        $P \leftarrow (P \setminus \{Z \in P \mid X' \succeq Z\}) \cup \{X'\}$;

    $t = t + 1$;

return $\arg\max_{X \in P : \hat{c}(X) \leq B} f(x)$

$$f_1(X) = \begin{cases} -\infty, & \hat{c}(X) > B + 1 \\ f(X), & \text{otherwise} \end{cases}, \quad f_2(X) = -\hat{c}(X).$$

Theorem 1

Generalized greedy algorithm is $\phi = (\alpha_f/2)(1 - \frac{1}{e^{\alpha_f}})$ approximation

Theorem 2

POMC algorithm is $\phi = (\alpha_f/2)(1 - \frac{1}{e^{\alpha_f}})$ approximation

Lemma 1

令 $v^* = \arg\max_{v \notin X} \frac{f(X \cup \{v\}) - f(X)}{\hat{c}(X \cup \{v\}) - \hat{c}(X)}$

有 $f(X \cup \{v^*\}) - f(X) \geq \alpha_f \dfrac{\hat{c}(X \cup \{v^*\}) - \hat{c}(X)}{B} \cdot (f(\tilde{X}) - f(X)).$

Proof

采用数学归纳法

设对于任意 b，在 P 中存在解 X 且存在正整数 i 满足 $\hat{c}(X) \le i < b$ 和 $f(X) \ge \left(1 - \left(1 - \alpha_f \frac{i}{bk}\right)^k\right) \cdot f(\tilde{X}_b)$

对于初始解 $X = \{0\}^n$ 和 i = 0 显然成立

满足定义的最大的 i 在算法执行过程中永远不会变小：

Case1：之前的 X 仍然在 P 中，i 依然成立
Case2：之前的 X 被替换，$\hat{c}(X)$ 变小 $f(X)$ 变大，i 依然成立

考虑一种使 i 增大的特殊情况：X 中加入了 v^*

Proof

$$f(X \cup \{v^*\}) - f(X) \geq \alpha_f \frac{\hat{c}(X \cup \{v^*\}) - \hat{c}(X)}{B} \cdot (f(\tilde{X}) - f(X)). \qquad (1)$$

$$f(X) \geq \left(1 - \left(1 - \alpha_f \frac{i}{bk}\right)^k\right) \cdot f(\tilde{X}_b) \qquad (2)$$

$$f(\tilde{X}) - f(X) \geq (1 - \alpha_f \frac{i}{Bk})^k f(\tilde{X})$$

$$f(\tilde{X}) - f(X \cup \{v^*\}) \geq (1 - \alpha_f \frac{i}{Bk})^k (1 - \alpha_f \frac{\hat{c}(X \cup \{v^*\}) - \hat{c}(X)}{B}) f(\tilde{X})$$

Proof

$$f(X \cup \{v^*\}) \geq \left(1 - \left(1 - \alpha_f \frac{b}{bk^*}\right)^{k^*}\right) \cdot f(\tilde{X}_b)$$
$$\geq \left(1 - \frac{1}{e^{\alpha_f}}\right) \cdot f(\tilde{X}_b).$$

$$f(X \cup \{v^*\}) \leq (f(X) + f(z))/\alpha_f.$$

$$\max\{f(X), f(z)\} \geq (\alpha_f/2) \cdot (1 - \frac{1}{e^{\alpha_f}}) \cdot f(\tilde{X}_b).$$

Proof

考虑概率 $\Pr(Y_j = 1) \geq \frac{1}{en} \cdot \frac{1}{P_{\max}}$

$T = cnP_{\max} \cdot \frac{B}{\delta_{\hat{c}}}$

$$E[Y^*] = \frac{T}{enP_{\max}} = \frac{cB}{e\delta_{\hat{c}}} \geq \frac{cb}{e\delta_{\hat{c}}}.$$

根据 Chernoff bound 取 $\delta = (1 - \frac{e}{c})$,

$$\Pr\left(Y < (1-\delta)E[Y^*]\right) \leq \Pr\left(Y^* < (1-\delta)E[Y^*]\right) \leq e^{-E[Y^*]\delta^2/2}.$$

$$\Pr(Y \leq \frac{b}{\delta_{\hat{c}}}) \leq e^{-\frac{(c-e)^2 B}{2ce\delta_{\hat{c}}}} \leq e^{-\frac{(c/2)^2 B}{2ce\delta_{\hat{c}}}} \leq e^{-\frac{cB}{8e\delta_{\hat{c}}}} \leq e^{-\frac{B}{\delta_{\hat{c}}}}.$$

Proof

重新定义 I，使其满足

$$f(X') \geq \left(1 - \left(1 - \alpha_f \frac{I_{\max} + \hat{c}(X') - \hat{c}(X)}{Bk}\right)^k\right) \cdot f(\tilde{X}_B)$$

and

$$f(X') \geq \left(1 - \left(1 - \alpha_f \frac{I_{\max} + \hat{c}(X') - \hat{c}(X)}{b'k'}\right)^{k'}\right) \cdot f(\tilde{X}_{b'}).$$

重新使用归纳法进行证明

对于在界限 B 下求得的解 X，在更大的界限 b 下，性质同样成立

Proof

Case1：若 $I_{\max} + \hat{c}(X') - \hat{c}(X) \geq b'$，则直接有 $\max\{f(X), f(z)\} \geq (\alpha_f/2) \cdot \left(1 - \frac{1}{e^{\alpha_f}}\right) \cdot f(\tilde{X}_{b'})$.

Case2：否则有 $f(X') \geq \left(1 - \left(1 - \alpha_f \frac{B}{b'k'}\right)^{k'}\right) \cdot f(\tilde{X}_{b'})$. 令之前证明中的 i = B，继续进行归纳证明

对于结果的概率分析同之前相同

Proof

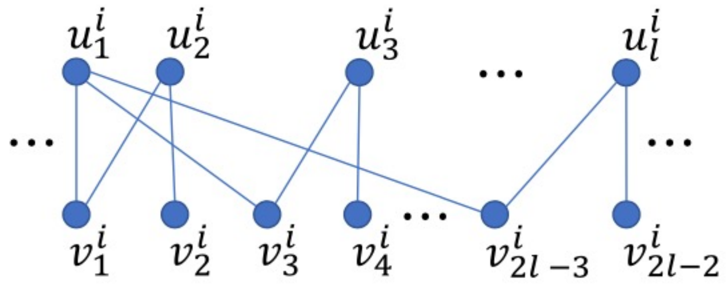
分析 adGCA 算法

B 增加时 approximation ratio 退化为 O(1/n)：

Given $n+1$ items $e_i = (c_i, f_i)$ with cost c_i and value f_i independent of the choice of the other items, we assume there are items $e_i = (1, \frac{1}{n})$, $1 \le i \le n/2$, $e_i = (2, 1)$, $n/2+1 \le i \le n$, and a special item $e_{n+1} = (1, 3)$. We have $f_{inc}(X) = \sum_{e_i \in X} f_i$ and $c_{inc}(X) = \sum_{e_i \in X} c_i$ as the linear objective and constraint function, respectively.

B 减小时 approximation ratio 退化为 O(1/√n):

构造 √n 个如下子图



Experiment

Problem

1. The influence maximization problem

2. The maximum coverage problem

Cost fuctions

1. routing constraint

2. cardinality constraint

Changing frequency τ

100, 1000, 5000, 10000

Experiment

1. GCA 表现普遍比 AdGCA 优
2. EAMC 算法比贪心劣
3. POMC 算法与 NSGA 算法表现接近
4. 在动态问题中 warming up 有明显的优化效果
5. τ 较小时贪心较优，τ 较大时进化算法较优
6. 在 cardinality constraint 下贪心优势更明显
7. POMC 中 P 的规模与 B 有一定关系
8. 在 POMC 算法执行一段时间后 P 会减小，找到了优秀的个体，降低了 P 规模；在此之后，基于这些优秀的个体，又能更容易的找到新解，使 P 规模上升

感谢观看

黄道煜

8208200412@csu.edu.cn

2022 年 5 月 13 日