オンライン最適化の競合比解析

東京大学 河瀬康志

組合せ最適化セミナー (第 18 回)
2021 年 8 月 4 日

アウトライン

オンライン最適化とは

- 逐次的な入力に対して逐次的に意思決定する状況での最適化

  将来のわからない状況で意思決定をする状況のモデル化

- 目標は最終的に得られる利得 (コスト) の最大化 (最小化)

  将来がわからない状況での最適化

- インターネットに関する最適化ではない

  オンラインショッピング，オンラインゲーム，オンラインバンキング，...

  ただし，オンライン最適化の枠組みで捉えられるインターネットに関する最適化もたくさんある

- ストリーミングアルゴリズムにかなり近いがコンセプトが違う

  ストリーミングでは全体の格納はできない大量データを逐次的に処理

  メモリが十分にない状況での最適化

オフライン vs オンライン

オフライン最適化

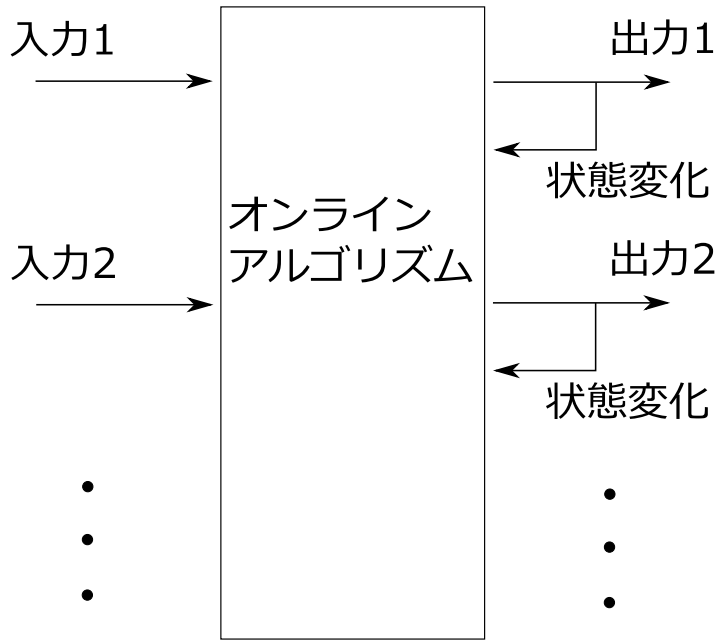
入力 一度に与えられる

目的 最適解を効率的に求める

オンライン最適化

入力 逐次的に与えられる

目的 なるべく良い選択



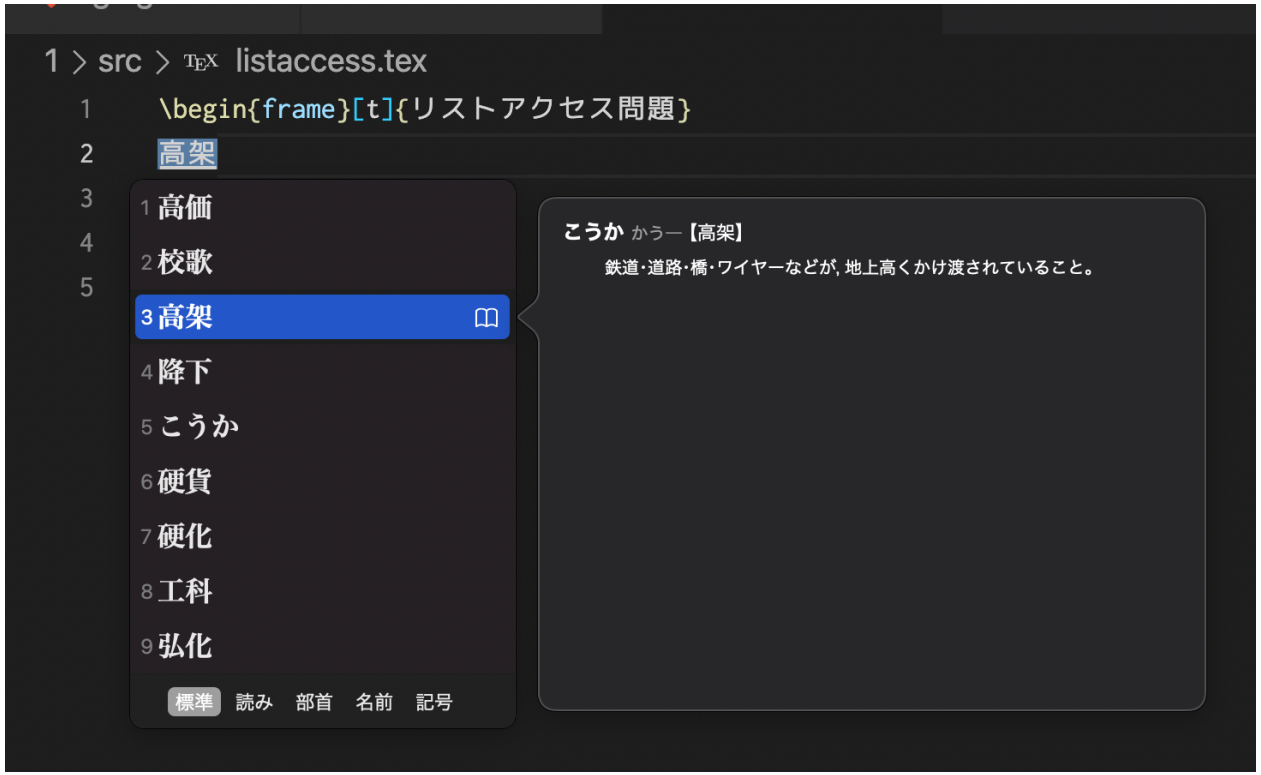
例 1: スキーレンタル問題

これからスキーを始めるが，今後何回スキーにいくかわからない．
スキー板を買うべきか，レンタルすべきか？

- レンタルすると，1 回につき 1 万円
- 買うと 5 万円
- 買ったスキーは，ずっと使い続けることができる（壊れない）

- 最初に買って 1 回しか行かなかったらもったいない
- 100 回行くのに毎回レンタルというのももったいない
- 何回行くかが分かればコスト最小化を計算できるが...

例 2: かな漢字変換 (リストアクセス問題)



- 過去の変換履歴から変換候補をうまく表示するプログラムを作りたい
- 変換候補をどのような順番で表示すれば効率よく変換できるか？
- 上の例の場合「降下」と変換するには 4 回リストを辿ることに

例 3: インターネット広告 (アドオークション問題)

https://search.yahoo.co.jp/

- ユーザーが来る度にどの広告を表示するか決める
- ユーザーの属性によって表示できる広告が異なる
- 各広告には表示回数の上限あり．沢山表示するにはどうする？

性能指標

オフライン最適化

入力 一度に与えられる

目的 最適解を効率的に求める

$$\text{近似比} := \frac{\mathrm{obj}(\mathrm{ALG}(I))}{\mathrm{obj}(\mathrm{OPT}(I))}$$

- 入力 I
- 多項式時間アルゴリズム ALG
- 最適アルゴリズム OPT

最適解を常に出力する

オンライン最適化

入力 逐次的に与えられる

目的 なるべく良い選択

$$\text{競合比} := \frac{\mathrm{obj}(\mathrm{ALG}(\sigma))}{\mathrm{obj}(\mathrm{OPT}(\sigma))}$$

- 入力列 σ
- オンラインアルゴリズム ALG
- 最適アルゴリズムOPT

最適オフラインアルゴリズム or 後出しオンラインアルゴリズム

オンラインアルゴリズムの性能指標

最小化オンライン問題に対する競合比 [Sleator and Tarjan 1985: Karlin et al. 1988.]

$$\sup_{\text{入力列}} \frac{\text{オンラインアルゴリズムでのコスト}}{\text{入力列を知っていた場合の最適コスト}}$$

最大化オンライン問題に対する競合比 逆数で定義する場合もあり

$$\sup_{\text{入力列}} \frac{\text{入力列を知っていた場合の最適利得}}{\text{オンラインアルゴリズムでの利得}}$$

コストや利得は常に非負と仮定．$0/0 = 1$ と仮定

- 最悪の場合に対し，最適コストの何倍以下に抑えられるかを保証
- 1 以上であり小さいほど嬉しい
- 競合比が小さいアルゴリズムが設計できると嬉しい
- 計算時間はとりあえず考えない 将来を知らないことによる影響だけに着目

注: 上の定義において競合比 ρ 以下であることを「競合比 ρ である」と言う場合もある

乱択オンラインアルゴリズムの性能指標

最小化オンライン問題に対する競合比 (oblivious adversary)

$$\sup_{\text{入力列}} \frac{\mathbb{E}[\text{乱択オンラインアルゴリズムでのコスト}]}{\text{入力列を知っていた場合の最適コスト}}$$

最大化オンライン問題に対する競合比

$$\sup_{\text{入力列}} \frac{\text{入力列を知っていた場合の最適利得}}{\mathbb{E}[\text{乱択オンラインアルゴリズムでの利得}]}$$

- 期待値の意味で最適コストの何倍以下に抑えられるかを保証
- 入力列は乱数に基づくアルゴリズムの挙動に依存しない
- 1 以上であり小さいほど嬉しい
- 競合比がなるべく小さいアルゴリズムを設計したい

注: オンライン最適化のリグレット解析

- 機械学習コミュニティなどでは，目的関数が未知の最適化問題を繰り返し解くことを「オンライン最適化」と呼ぶこともある

  エキスパート問題，多腕バンディット問題，オンライン線形最適化，オンライン凸最適化，...

- このような場合はリグレットを性能指標にすることが多い

$$\text{リグレット} = \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} f_t(x_t)\right]}_{\text{アルゴリズムの期待累積損失}} - \underbrace{\min_{x^* \in X} \sum_{t=1}^{T} f_t(x^*)}_{\text{最適累積損失}}$$

- 今日は繰り返し構造のないオンライン最適化のみを扱う

今日やること

「双対」の関係を用いたオンラインアルゴリズムの設計と競合比解析

- 主双対法に基づくオンラインアルゴリズムの設計

  参考文献 *Niv Buchbinder and Joseph (Seffi) Naor: The Design of Competitive Online Algorithms via a Primal-Dual Approach. Foundations and Trends® in Theoretical Computer Science, Vol. 3, Nos. 2–3 (2007), pp. 93–263.*

- Yao の原理を用いた不可能性の証明

復習: 線形計画問題

主問題

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

双対問題

$$\begin{aligned} \max \quad & b^\top y \\ \text{s.t.} \quad & A^\top y \leq c \\ & y \geq 0 \end{aligned}$$

弱双対性 任意の実行可能解 x, y について $c^\top x \geq b^\top y$

強双対性 最適解 x^*, y^* について $c^\top x^* = b^\top y^*$

相補性 実行可能解 x, y が最適解であるための必要十分条件は

- 任意の i について $x_i > 0 \Rightarrow \sum_{j=1}^{m} a_{ij} y_i = c_i$;
- 任意の j について $y_j > 0 \Rightarrow \sum_{i=1}^{n} a_{ij} x_i = b_j$

復習: 線形計画問題

主問題

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

双対問題

$$\begin{aligned} \max \quad & b^\top y \\ \text{s.t.} \quad & A^\top y \leq c \\ & y \geq 0 \end{aligned}$$

近似相補性条件

実行可能解 x, y が以下の条件を満たす $\Rightarrow$ $c^\top x \leq \alpha \cdot \beta b^\top y$

- 任意の i について $x_i > 0 \Rightarrow c_i/\alpha \leq \sum_{j=1}^{m} a_{ij} y_i \leq c_i$;
- 任意の j について $y_j > 0 \Rightarrow b_j \leq \sum_{i=1}^{n} a_{ij} x_i \leq \beta \cdot b_j$

証明

$$\sum_{i=1}^{n} c_i x_i \leq \sum_{i=1}^{n} \left(\alpha \sum_{j=1}^{m} a_{ij} y_j \right) x_i = \alpha \cdot \sum_{j=1}^{m} \left(\sum_{i=1}^{n} a_{ij} x_i \right) y_j \leq \alpha \cdot \beta \sum_{j=1}^{m} b_j y_j$$

線形計画の例

問題 P

$$
\begin{aligned}
\min \quad & 22x_1 + 11x_2 \\
\text{s.t.} \quad & 4x_1 + 5x_2 \geq 13 \\
& 3x_1 + x_2 \geq 7 \\
& 2x_1 + 6x_2 \geq 9 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

問題 D

$$
\begin{aligned}
\max \quad & 13y_1 + 7y_2 + 9y_3 \\
\text{s.t.} \quad & 4y_1 + 3y_2 + 2y_3 \leq 22 \\
& 5y_1 + y_2 + 6y_3 \leq 11 \\
& y_1, y_2, y_3 \geq 0
\end{aligned}
$$

- 問: 問題 P の最適値は 100 以下？

  解: Yes! $(x_1, x_2) = (1, 4)$ とすれば実行可能かつ目的関数値は 66

-

- 最適値は 55
  - P の最適解: $(x_1, x_2) = (2, 1)$
  - D の最適解: $(y_1, y_2, y_3) = (1, 6, 0)$
  - 強双対性，相補性条件の成立も確認できる

線形計画の例

問題 P

$$\begin{aligned} \min \quad & 22x_1 + 11x_2 \\ \text{s.t.} \quad & 4x_1 + 5x_2 \geq 13 \\ & 3x_1 + x_2 \geq 7 \\ & 2x_1 + 6x_2 \geq 9 \\ & x_1, x_2 \geq 0 \end{aligned}$$

問題 D

$$\begin{aligned} \max \quad & 13y_1 + 7y_2 + 9y_3 \\ \text{s.t.} \quad & 4y_1 + 3y_2 + 2y_3 \leq 22 \\ & 5y_1 + y_2 + 6y_3 \leq 11 \\ & y_1, y_2, y_3 \geq 0 \end{aligned}$$

- 問: 問題 P の最適値は 100 以下？

  解: Yes! $(x_1, x_2) = (1, 4)$ とすれば実行可能かつ目的関数値は 66

-
- 最適値は 55
  - P の最適解: $(x_1, x_2) = (2, 1)$
  - D の最適解: $(y_1, y_2, y_3) = (1, 6, 0)$
  - 強双対性，相補性条件の成立も確認できる

線形計画の例

問題 P

$$\begin{aligned} \min \quad & 22x_1 + 11x_2 \\ \text{s.t.} \quad & 4x_1 + 5x_2 \geq 13 \\ & 3x_1 + x_2 \geq 7 \\ & 2x_1 + 6x_2 \geq 9 \\ & x_1, x_2 \geq 0 \end{aligned}$$

問題 D

$$\begin{aligned} \max \quad & 13y_1 + 7y_2 + 9y_3 \\ \text{s.t.} \quad & 4y_1 + 3y_2 + 2y_3 \leq 22 \\ & 5y_1 + y_2 + 6y_3 \leq 11 \\ & y_1, y_2, y_3 \geq 0 \end{aligned}$$

- 問: 問題 P の最適値は 100 以下？

  解: Yes! $(x_1, x_2) = (1, 4)$ とすれば実行可能かつ目的関数値は 66

- 問: 問題 P の最適値は 30 以下？

- 最適値は 55
  - P の最適解: $(x_1, x_2) = (2, 1)$
  - D の最適解: $(y_1, y_2, y_3) = (1, 6, 0)$
  - 強双対性，相補性条件の成立も確認できる

線形計画の例

問題 P

$$
\begin{aligned}
\min \quad & 22x_1 + 11x_2 \\
\text{s.t.} \quad & 4x_1 + 5x_2 \geq 13 \\
& 3x_1 + x_2 \geq 7 \\
& 2x_1 + 6x_2 \geq 9 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

問題 D

$$
\begin{aligned}
\max \quad & 13y_1 + 7y_2 + 9y_3 \\
\text{s.t.} \quad & 4y_1 + 3y_2 + 2y_3 \leq 22 \\
& 5y_1 + y_2 + 6y_3 \leq 11 \\
& y_1, y_2, y_3 \geq 0
\end{aligned}
$$

- 問: 問題 P の最適値は 100 以下？

  解: Yes! $(x_1, x_2) = (1, 4)$ とすれば実行可能かつ目的関数値は 66

- 問: 問題 P の最適値は 30 以下？

  解: No! 全部の (x_1, x_2) を調べても 30 以下にはならない!?

- 最適値は 55
  - P の最適解: $(x_1, x_2) = (2, 1)$
  - D の最適解: $(y_1, y_2, y_3) = (1, 6, 0)$

線形計画の例

問題 P

$$\begin{aligned} \min \quad & 22x_1 + 11x_2 \\ \text{s.t.} \quad & 4x_1 + 5x_2 \geq 13 \\ & 3x_1 + x_2 \geq 7 \\ & 2x_1 + 6x_2 \geq 9 \\ & x_1, x_2 \geq 0 \end{aligned}$$

問題 D

$$\begin{aligned} \max \quad & 13y_1 + 7y_2 + 9y_3 \\ \text{s.t.} \quad & 4y_1 + 3y_2 + 2y_3 \leq 22 \\ & 5y_1 + y_2 + 6y_3 \leq 11 \\ & y_1, y_2, y_3 \geq 0 \end{aligned}$$

- 問: 問題 P の最適値は 100 以下？

  解: Yes! $(x_1, x_2) = (1, 4)$ とすれば実行可能かつ目的関数値は 66

- 問: 問題 P の最適値は 30 以下？

  解: No! 全部の (x_1, x_2) を調べても 30 以下にはならない!?

  $(y_1, y_2, y_3) = (2, 1, 0)$ とすれば実行可能かつ目的関数値は 33

  弱双対性により No の証拠となる

  $(x_1, x_2) = (1, 4)$ が 2 近似 (よりも悪くない) 解である証拠にもなる

線形計画の例

問題 P

$$
\begin{aligned}
\min \quad & 22x_1 + 11x_2 \\
\text{s.t.} \quad & 4x_1 + 5x_2 \geq 13 \\
& 3x_1 + x_2 \geq 7 \\
& 2x_1 + 6x_2 \geq 9 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

問題 D

$$
\begin{aligned}
\max \quad & 13y_1 + 7y_2 + 9y_3 \\
\text{s.t.} \quad & 4y_1 + 3y_2 + 2y_3 \leq 22 \\
& 5y_1 + y_2 + 6y_3 \leq 11 \\
& y_1, y_2, y_3 \geq 0
\end{aligned}
$$

- 最適値は 55
  - P の最適解: $(x_1, x_2) = (2, 1)$
  - D の最適解: $(y_1, y_2, y_3) = (1, 6, 0)$
  - 強双対性，相補性条件の成立も確認できる

アウトライン

スキーレンタル問題

これからスキーを始めるが，今後何回スキーにいくかわからない．
スキー板を買うべきか，レンタルすべきか？

- レンタルすると，1 回につき 1 万円
- 買うと 5 万円
- 買ったスキーは，ずっと使い続けることができる（壊れない）

スキーレンタル問題

これからスキーを始めるが，今後何回スキーにいくかわからない．
スキー板を買うべきか，レンタルすべきか？

- レンタルすると，1 回につき 1 万円
- 買うと 5 万円
- 買ったスキーは，ずっと使い続けることができる（壊れない）

アルゴリズム A: ずっとレンタルし続ける

- 100 回行くと 100 万円かかるが，最初に購入しておけば 5 万円
  → 競合比は $100/5 = 20$ 以上
- k (≥ 5) 回行くとすると，競合比は $k/5 \to \infty$ $(k \to \infty)$

スキーレンタル問題

これからスキーを始めるが，今後何回スキーにいくかわからない．
スキー板を買うべきか，レンタルすべきか？

- レンタルすると，1 回につき 1 万円
- 買うと 5 万円
- 買ったスキーは，ずっと使い続けることができる（壊れない）

アルゴリズム A: ずっとレンタルし続ける

- 100 回行くと 100 万円かかるが，最初に購入しておけば 5 万円
  → 競合比は $100/5 = 20$ 以上
- k (≥ 5) 回行くとすると，競合比は $k/5 \to \infty$ $(k \to \infty)$

アルゴリズム B: 最初に購入する

- 1 回しか行かない場合が最悪で，競合比は $5/1 = 5$

スキーレンタル問題

これからスキーを始めるが，今後何回スキーにいくかわからない．
スキー板を買うべきか，レンタルすべきか？

- レンタルすると，1 回につき 1 万円
- 買うと 5 万円
- 買ったスキーは，ずっと使い続けることができる（壊れない）

アルゴリズム C: 4 回目まではレンタルで 5 回目に購入する

スキーに k 回行く場合：

- $k \leq 4$ のとき
  毎回レンタルが最適であり，競合比は 1
- $k \geq 5$ のとき
  最初に購入が最適であり，競合比は $(4+5)/5 = 1.8$

スキーレンタル問題

これからスキーを始めるが，今後何回スキーにいくかわからない．
スキー板を買うべきか，レンタルすべきか？

- レンタルすると，1 回につき 1 万円
- 買うと 5 万円
- 買ったスキーは，ずっと使い続けることができる（壊れない）

もっと良いアルゴリズムはあるか？ → 決定的な範囲にはない

スキーレンタル問題

これからスキーを始めるが，今後何回スキーにいくかわからない．
スキー板を買うべきか，レンタルすべきか？

- レンタルすると，1 回につき 1 万円
- 買うと 5 万円
- 買ったスキーは，ずっと使い続けることができる（壊れない）

もっと良いアルゴリズムはあるか？ → 決定的な範囲にはない

- 一度購入したらレンタルをすることはないので，
「何回目に買うか」でアルゴリズムは決まる
- ℓ 回目に買うアルゴリズムを考える
- 最悪なのは ℓ 回しかスキーに行かない場合であり，競合比は

$$\frac{(\ell - 1) + 5}{\min\{\ell, 5\}} \geq \frac{9}{5} = 1.8$$

スキーレンタル問題 (一般の形)

これからスキーを始めるが，今後何回スキーにいくかわからない．
スキー板を買うべきか，レンタルすべきか？

- レンタルすると，1 回につき 1 万円
- 買うと B 万円（B は整数とする）
- 買ったスキーは，ずっと使い続けることができる（壊れない）

- 最適なオンラインアルゴリズムは $B-1$ 回のレンタル後に購入
- 最悪の状況は B 回スキーに行く場合で，競合比は $\frac{(B-1)+B}{B} = 2 - \frac{1}{B}$

整数計画問題によるスキーレンタル問題の定式化

$$
\begin{aligned}
\min \quad & B \cdot x + \textstyle\sum_{j=1}^{k} z_j & \\
\text{s.t.} \quad & x + z_j \geq 1 & (\forall j \in \{1, 2, \ldots, k\}) \\
& x \in \{0, 1\}, z_j \in \{0, 1\} & (\forall j \in \{1, 2, \ldots, k\})
\end{aligned}
$$

- x: 購入するかどうかを表す 0–1 変数
- z_j: j 回目にレンタルするかどうかを表す 0–1 変数

ただし j 回目に関する変数・制約は j 回目までわからない j 回目は来ないかも

1 回目

$$
\begin{aligned}
\min \quad & B \cdot x + z_1 \\
\text{s.t.} \quad & x + z_1 \geq 1 \\
& x, z_1 \in \{0, 1\}
\end{aligned}
$$

2 回目

$$
\begin{aligned}
\min \quad & B \cdot x + z_1 + z_2 \\
\text{s.t.} \quad & x + z_1 \geq 1 \\
& x + z_2 \geq 1 \\
& x, z_1, z_2 \in \{0, 1\}
\end{aligned}
$$

3 回目

$$
\begin{aligned}
\min \quad & B \cdot x + z_1 + z_2 + z_3 \\
\text{s.t.} \quad & x + z_1 \geq 1 \\
& x + z_2 \geq 1 \\
& x + z_3 \geq 1 \\
& x, z_1, z_2, z_3 \in \{0, 1\}
\end{aligned}
$$

整数計画問題によるスキーレンタル問題の定式化

$$
\begin{aligned}
\min \quad & B \cdot x + \sum_{j=1}^{k} z_j & \\
\text{s.t.} \quad & x + z_j \geq 1 & (\forall j \in \{1, 2, \ldots, k\}) \\
& x \in \{0, 1\}, z_j \in \{0, 1\} & (\forall j \in \{1, 2, \ldots, k\})
\end{aligned}
$$

- x: 購入するかどうかを表す 0–1 変数
- z_j: j 回目にレンタルするかどうかを表す 0–1 変数

- 各ラウンド j では j に関する制約がくる
- これまでにきた制約を全て満たす必要がある
- 各ラウンドでは変数を増やすことしかできない
  過去の購入 or レンタルの決定を後から覆すことはできない

線形緩和と双対問題

主問題

$$\begin{aligned} \min \quad & B \cdot x + \textstyle\sum_{j=1}^{k} z_j & \\ \text{s.t.} \quad & x + z_j \geq 1 & (\forall j) \\ & x, z_j \geq 0 & (\forall j) \end{aligned}$$

双対問題

$$\begin{aligned} \max \quad & \textstyle\sum_{j=1}^{k} y_j & \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{k} y_j \leq B & \\ & 0 \leq y_j \leq 1 & (\forall j) \end{aligned}$$

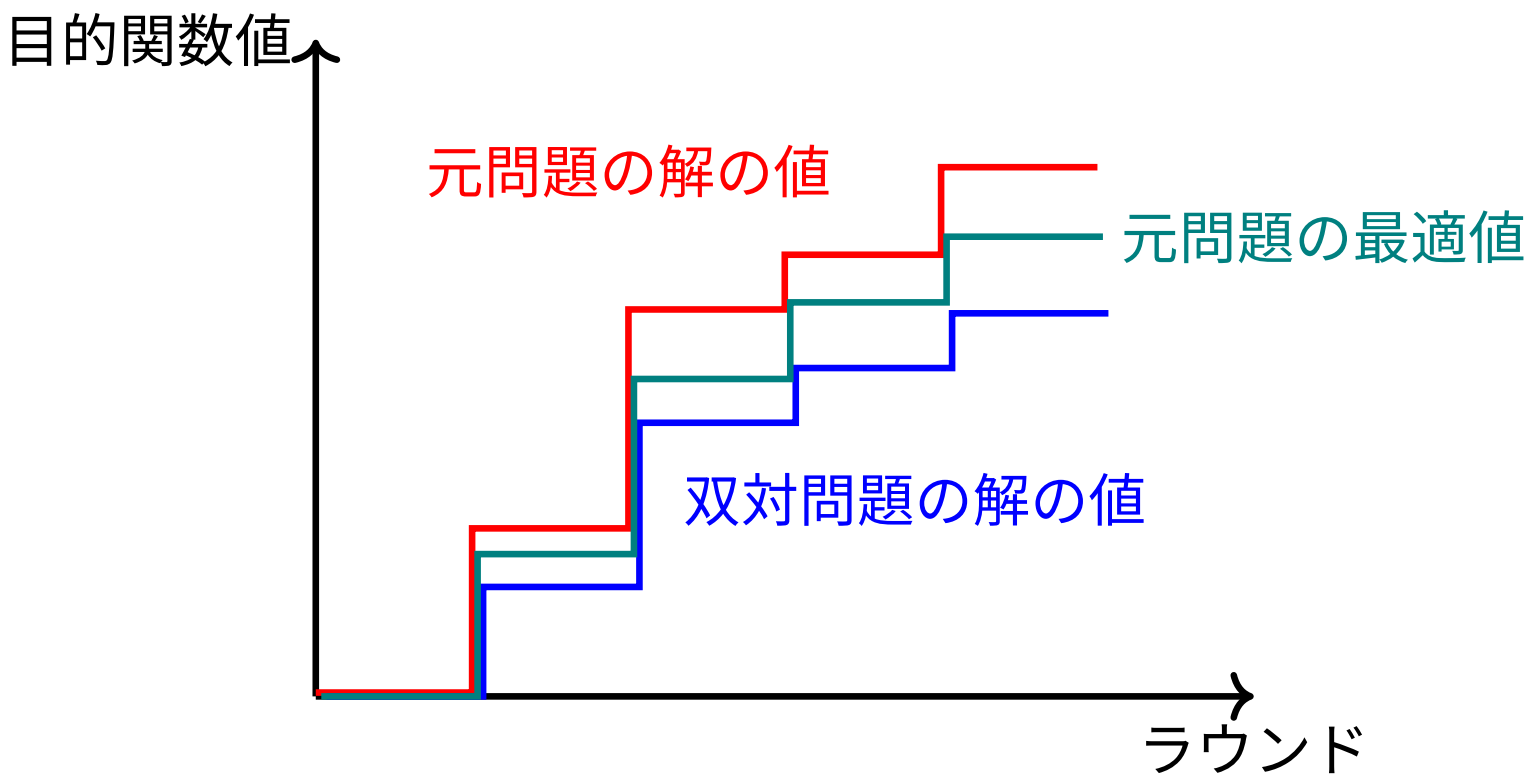
- x は購入確率，z_j は j 回目にレンタルする確率と解釈可能
- d を $[0, 1]$ 上の一様乱数として，$x \geq d$ となれば購入するとすれば，実行可能であり，期待コストは $Bx + \sum_{j=1}^{k} z_j$ となる

オンライン主双対法のアイデア

- 元問題の解 と 双対問題の実行可能解 を同時に構成
- 各ラウンドでの目的関数値の変化の比を ρ で抑える

  → 最終的な値の比も ρ 以下 → アルゴリズムは競合比 ρ

弱双対性より，主問題の整数解の値 $\geq$ 主問題の最適値 $\geq$ 双対問題の解の値



オンライン主双対法に基づくアルゴリズム

主問題

$$\begin{aligned} \min \quad & B \cdot x + \sum_{j=1}^{k} z_j & \\ \text{s.t.} \quad & x + z_j \geq 1 & (\forall j) \\ & x, z_j \geq 0 & (\forall j) \end{aligned}$$

双対問題

$$\begin{aligned} \max \quad & \sum_{j=1}^{k} y_j & \\ \text{s.t.} \quad & \sum_{j=1}^{k} y_j \leq B & \\ & 0 \leq y_j \leq 1 & (\forall j) \end{aligned}$$

初期化: $x \leftarrow 0$;

Foreach j 回目:

if $x < 1$:

(a) $z_j \leftarrow 1 - x$;
(b) $x \leftarrow x(1 + 1/B) + 1/(c \cdot B)$ (c は後で決める);
(c) $y_j \leftarrow 1$;

else: $z_j \leftarrow 0,\ y_j \leftarrow 0$;

アルゴリズムの解析

次の 3 つが成り立つ $\Rightarrow$ 競合比 $(1 + 1/c)$

1. 主問題について常に実行可能
2. 双対問題について常に実行可能
3. $\Delta P_j \leq (1 + \frac{1}{c})\Delta D_j$

初期化: $x \leftarrow 0$;
Foreach j 回目:
  if $x < 1$:
    (a) $z_j \leftarrow 1 - x$;
    (b) $x \leftarrow x(1 + \frac{1}{B}) + \frac{1}{c \cdot B}$;
    (c) $y_j \leftarrow 1$;
  else: $z_j \leftarrow 0,\ y_j \leftarrow 0$;

アルゴリズムの解析

次の 3 つが成り立つ $\Rightarrow$ 競合比 $(1+1/c)$

1. 主問題について常に実行可能
2. 双対問題について常に実行可能
3. $\Delta P_j \leq (1+\frac{1}{c})\Delta D_j$

初期化: $x \leftarrow 0$;
Foreach j 回目:
  if $x < 1$:
    (a) $z_j \leftarrow 1 - x$;
    (b) $x \leftarrow x(1+\frac{1}{B}) + \frac{1}{c \cdot B}$;
    (c) $y_j \leftarrow 1$;
  else: $z_j \leftarrow 0$, $y_j \leftarrow 0$;

1. の成立

- $x \geq 1$ なら明らかに実行可能
- $x < 1$ の場合は，
  $z_j \leftarrow 1 - x$ と設定していることと
  x は単調増加であることから実行可能

主問題

$$\begin{aligned} \min \quad & B \cdot x + \textstyle\sum_{j=1}^{k} z_j & \\ \text{s.t.} \quad & x + z_j \geq 1 & (\forall j) \\ & x, z_j \geq 0 & (\forall j) \end{aligned}$$

アルゴリズムの解析

次の 3 つが成り立つ $\Rightarrow$ 競合比 $(1+1/c)$

1. 主問題について常に実行可能
2. 双対問題について常に実行可能
3. $\Delta P_j \leq (1+\frac{1}{c})\Delta D_j$

初期化: $x \leftarrow 0$;
Foreach j 回目:
  if $x < 1$:
    (a) $z_j \leftarrow 1-x$;
    (b) $x \leftarrow x(1+\frac{1}{B})+\frac{1}{c\cdot B}$;
    (c) $y_j \leftarrow 1$;
  else: $z_j \leftarrow 0,\ y_j \leftarrow 0$;

2. の成立

- B 回目に $x \geq 1$ となれば良い
- B 回目の開始時に $x < 1$ の時
  $x = \sum_{j=1}^{B} \frac{1}{cB}\cdot(1+\frac{1}{B})^{j-1} = \frac{(1+\frac{1}{B})^B-1}{c}$
- $c \leq (1+\frac{1}{B})^B - 1$ なら成立
  $(1+\frac{1}{c} = \frac{(1+\frac{1}{B})^B}{(1+\frac{1}{B})^B-1} \approx \frac{e}{e-1})$

双対問題

$$\begin{aligned} \max \quad & \textstyle\sum_{j=1}^{k} y_j \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{k} y_j \leq B \\ & 0 \leq y_j \leq 1 \qquad (\forall j) \end{aligned}$$

アルゴリズムの解析

次の 3 つが成り立つ $\Rightarrow$ 競合比 $(1 + 1/c)$

1. 主問題について常に実行可能
2. 双対問題について常に実行可能
3. $\Delta P_j \leq (1 + \frac{1}{c})\Delta D_j$

初期化: $x \leftarrow 0$;
Foreach j 回目:
  if $x < 1$:
    (a) $z_j \leftarrow 1 - x$;
    (b) $x \leftarrow x(1 + \frac{1}{B}) + \frac{1}{c \cdot B}$;
    (c) $y_j \leftarrow 1$;
  else: $z_j \leftarrow 0,\ y_j \leftarrow 0$;

3. の成立

- $x \geq 1$ の時，$\Delta P_j = \Delta D_j = 0$
- $x < 1$ の時，
  - $\Delta P_j = \Delta(B \cdot x + \sum_{j'=1}^{j} z_{j'}) = B \cdot (\frac{x}{B} + \frac{1}{c \cdot B}) + 1 - x = 1 + \frac{1}{c}$
  - $\Delta D_j = \Delta(\sum_{j'=1}^{j} y_j) = 1$

スキーレンタル問題まとめ

定理

主双対法に基づくアルゴリズムの競合比は $1 + \frac{1}{c} = \frac{(1+\frac{1}{B})^B}{(1+\frac{1}{B})^B-1} \approx \frac{e}{e-1}$

$(1+\frac{1}{B})^B - 1$

アルゴリズム

初期化: $x \leftarrow 0$, $d \leftarrow [0,1]$ 上の一様乱数;
Foreach j 回目:
  if $x < 1$:
    (a) $z_j \leftarrow 1 - x$;
    (b) $x \leftarrow x(1 + \frac{1}{B}) + \frac{1}{c \cdot B}$;
    (c) $y_j \leftarrow 1$;
  else: $z_j \leftarrow 0$, $y_j \leftarrow 0$;

$x \geq d$ かつまだスキーを購入していなければ購入，$x < d$ ならばレンタル

これよりも競合比の意味で良いアルゴリズムは存在しない
全ての乱択アルゴリズムを考えることで Yao の原理により証明可能

スキーレンタル問題まとめ

定理

主双対法に基づくアルゴリズムの競合比は $1 + \frac{1}{c} = \frac{(1+\frac{1}{B})^B}{(1+\frac{1}{B})^B-1} \approx \frac{e}{e-1}$

$c = (1+\frac{1}{B})^B - 1$

アルゴリズム

初期化: $x \leftarrow 0$, $d \leftarrow [0,1]$ 上の一様乱数;
Foreach j 回目:
  if $x < 1$:
    (a) $z_j \leftarrow 1 - x$;
    (b) $x \leftarrow x(1 + \frac{1}{B}) + \frac{1}{c \cdot B}$;
    (c) $y_j \leftarrow 1$;
  else: $z_j \leftarrow 0$, $y_j \leftarrow 0$;

$x \geq d$ かつまだスキーを購入していなければ購入，$x < d$ ならばレンタル

これよりも競合比の意味で良いアルゴリズムは存在しない
~~全ての乱択アルゴリズムを考えることで~~ Yao の原理により証明可能

アウトライン

準備: 乱択アルゴリズムの表現

乱択アルゴリズムは決定性アルゴリズム上の確率分布として表現できる

- スキーレンタル問題に対する乱択アルゴリズムでは，
  何回目に購入するかを最初に決定（k 回目に購入する確率 p_k）
  → k 回目に購入という決定性アルゴリズムを確率 p_k で選択している
- しかし，一般には入力に依存してランダムな挙動を取るかもしれない

  例: リストアクセス問題において変換履歴に応じて確率的に順序を定めるなど

- このような場合も，途中で利用しうる全ての乱数を $\boldsymbol{r}$ とし
  これを引数とするアルゴリズムと考えることで，
  決定的なアルゴリズム $\mathrm{ALG}(\sigma, \boldsymbol{r})$ に変換できる

ゲーム理論的解釈

我々 vs. 敵

最悪ケース解析は二人零和ゲームだと解釈可能

- 我々の戦略は決定性アルゴリズム 混合戦略が乱択アルゴリズム
- 敵の戦略は入力 本当はあらゆる入力を考えているだけだが，敵が一番悪い入力を選ぶと想定する
- 2 人の戦略が決まると「コスト」が決定される コストは目的関数値，競合比，計算時間，など

例: スキーレンタル問題 $(N = 5)$ のコスト行列 (コストは目的関数値)

敵＼我々	1	2	3	4	5	6	...
1	5	1	1	1	1	1	...
2	5	6	2	2	2	2	...
3	5	6	7	3	3	3	...
4	5	6	7	8	4	4	...
5	5	6	7	8	9	5	...
6	5	6	7	8	9	10	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

最適な乱択アルゴリズム

最適乱択アルゴリズム　最悪の入力

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \mathbb{E}_{y\sim\boldsymbol{q}}[\mathrm{cost}(\mathrm{ALG}_y(\sigma))]$$

- 最悪の入力に対して「コスト」の小さい乱択アルゴリズムを設計したい
- 乱択アルゴリズムを選択した後に敵が最悪の入力を選択する

後手は決定的でも有利 (弱双対性)

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \mathbb{E}_{y\sim\boldsymbol{q}}[\mathrm{cost}(\mathrm{ALG}_y(\sigma))] \geq \sup_{\boldsymbol{p}} \inf_{\mathrm{ALG}} \mathbb{E}_{x\sim\boldsymbol{p}}[\mathrm{cost}(\mathrm{ALG}(\sigma_x))]$$

- 多くの場合等号が成立
- 等号が成り立たなくても，右辺により
  あるコスト以下の乱択アルゴリズムが存在しない証拠を与えられる

Yao の原理 (基本形)

後手は決定的でも有利 (弱双対性)

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \mathbb{E}_{y\sim\boldsymbol{q}}[\mathrm{cost}(\mathrm{ALG}_y(\sigma))] \geq \sup_{\boldsymbol{p}} \inf_{\mathrm{ALG}} \mathbb{E}_{x\sim\boldsymbol{p}}[\mathrm{cost}(\mathrm{ALG}(\sigma_x))]$$

Andrew Chi-Chih Yao: 2000 年 Turing 賞，2021 年京都賞

Yao の原理 (基本形)

任意の入力の分布 $\hat{\boldsymbol{p}}$ について

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \mathbb{E}_{y\sim\boldsymbol{q}}[\mathrm{cost}(\mathrm{ALG}_y(\sigma))] \geq \inf_{\mathrm{ALG}} \mathbb{E}_{x\sim\hat{\boldsymbol{p}}}[\mathrm{cost}(\mathrm{ALG}(\sigma_x))]$$

- 乱択アルゴリズムに対する不可能性を示す基本テクニック
- 右辺により最適乱択アルゴリズムに対するコストの下界が得られる
- 我々と敵（アルゴリズムと入力）の立場が入れ替わり，
  どんなアルゴリズムに対してもコストが大きい乱択入力の設計問題に

競合比に対する Yao の原理（最小化問題）

$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathbb{E}_{y\sim \boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]}{\mathrm{obj}(\mathrm{OPT}(\sigma))}$ を評価したい

$\mathrm{cost}(\mathrm{ALG}(\sigma)) = \frac{\mathrm{obj}(\mathrm{ALG}(\sigma))}{\mathrm{obj}(\mathrm{OPT}(\sigma))}$ として Yao の原理を用いると次が成立

競合比に対する Yao の原理 1

任意の入力の分布 $\hat{\boldsymbol{p}}$ について

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathbb{E}_{y\sim \boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]}{\mathrm{obj}(\mathrm{OPT}(\sigma))} \geq \inf_{\mathrm{ALG}} \mathbb{E}_{x\sim \hat{\boldsymbol{p}}} \left[\frac{\mathrm{obj}(\mathrm{ALG}(\sigma_x))}{\mathrm{obj}(\mathrm{OPT}(\sigma_x))}\right]$$

$\inf_q \sup_\sigma \mathbb{E}_{y\sim q}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]/\mathrm{obj}(\mathrm{OPT}(\sigma))$

$\mathrm{cost}(\mathrm{ALG}(\sigma)) = \mathrm{obj}(\mathrm{ALG}(\sigma)) - c \cdot \mathrm{obj}(\mathrm{OPT}(\sigma))$ とすると以下を示せる

競合比に対する Yao の原理 2

任意の入力の分布 $\hat{\boldsymbol{p}}$ について

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathbb{E}_{y\sim \boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]}{\mathrm{obj}(\mathrm{OPT}(\sigma))} \geq \inf_{\mathrm{ALG}} \frac{\mathbb{E}_{x\sim \hat{\boldsymbol{p}}}[\mathrm{obj}(\mathrm{ALG}(\sigma_x))]}{\mathbb{E}_{x\sim \hat{\boldsymbol{p}}}[\mathrm{obj}(\mathrm{OPT}(\sigma_x))]}$$

競合比に対する Yao の原理（最小化問題）

$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathbb{E}_{y\sim \boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]}{\mathrm{obj}(\mathrm{OPT}(\sigma))}$ を評価したい

$\mathrm{cost}(\mathrm{ALG}(\sigma)) = \frac{\mathrm{obj}(\mathrm{ALG}(\sigma))}{\mathrm{obj}(\mathrm{OPT}(\sigma))}$ として Yao の原理を用いると次が成立

競合比に対する Yao の原理 1

任意の入力の分布 $\hat{\boldsymbol{p}}$ について

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathbb{E}_{y\sim \boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]}{\mathrm{obj}(\mathrm{OPT}(\sigma))} \geq \inf_{\mathrm{ALG}} \mathbb{E}_{x\sim \hat{\boldsymbol{p}}} \left[\frac{\mathrm{obj}(\mathrm{ALG}(\sigma_x))}{\mathrm{obj}(\mathrm{OPT}(\sigma_x))}\right]$$

$\inf_{\boldsymbol{q}} \sup_{\sigma} \mathbb{E}_{y\sim \boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]/\mathrm{obj}(\mathrm{OPT}(\sigma))$

$\mathrm{cost}(\mathrm{ALG}(\sigma)) = \mathrm{obj}(\mathrm{ALG}(\sigma)) - c \cdot \mathrm{obj}(\mathrm{OPT}(\sigma))$ とすると以下を示せる

競合比に対する Yao の原理 2

任意の入力の分布 $\hat{\boldsymbol{p}}$ について

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathbb{E}_{y\sim \boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]}{\mathrm{obj}(\mathrm{OPT}(\sigma))} \geq \inf_{\mathrm{ALG}} \frac{\mathbb{E}_{x\sim \hat{\boldsymbol{p}}}[\mathrm{obj}(\mathrm{ALG}(\sigma_x))]}{\mathbb{E}_{x\sim \hat{\boldsymbol{p}}}[\mathrm{obj}(\mathrm{OPT}(\sigma_x))]}$$

競合比に対する Yao の原理（最大化問題）

$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathrm{obj}(\mathrm{OPT}(\sigma))}{\mathbb{E}_{y\sim\boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]}$ を評価したい

競合比に対する Yao の原理 1

任意の入力の分布 $\hat{\boldsymbol{p}}$ について

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathrm{obj}(\mathrm{OPT}(\sigma))}{\mathbb{E}_{y\sim\boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]} \geq \inf_{\mathrm{ALG}} \frac{1}{\mathbb{E}_{x\sim\hat{\boldsymbol{p}}}\left[\frac{\mathrm{obj}(\mathrm{ALG}(\sigma_x))}{\mathrm{obj}(\mathrm{OPT}(\sigma_x))}\right]}$$

$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathrm{obj}(\mathrm{OPT}(\sigma))}{\mathbb{E}_{y\sim\boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]} \geq \inf_{\mathrm{ALG}} \mathbb{E}_{x\sim\hat{\boldsymbol{p}}}\left[\frac{\mathrm{obj}(\mathrm{OPT}(\sigma_x))}{\mathrm{obj}(\mathrm{ALG}(\sigma_x))}\right]$ は成り立たない

競合比に対する Yao の原理 2

任意の入力の分布 $\hat{\boldsymbol{p}}$ について

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \frac{\mathrm{obj}(\mathrm{OPT}(\sigma))}{\mathbb{E}_{y\sim\boldsymbol{q}}[\mathrm{obj}(\mathrm{ALG}_y(\sigma))]} \geq \inf_{\mathrm{ALG}} \frac{\mathbb{E}_{x\sim\hat{\boldsymbol{p}}}[\mathrm{obj}(\mathrm{OPT}(\sigma_x))]}{\mathbb{E}_{x\sim\hat{\boldsymbol{p}}}[\mathrm{obj}(\mathrm{ALG}(\sigma_x))]}$$

スキーレンタル問題の乱択競合比下界

入力の分布：ちょうど k 回行く確率が $\frac{1}{B} \cdot \left(1 - \frac{1}{B}\right)^{k-1}$

$\sum_{k=1}^{\infty} \frac{1}{B} \cdot \left(1 - \frac{1}{B}\right)^{k-1} = 1$

- 最適コストの期待値 $= B\left(1 - \left(1 - \frac{1}{B}\right)^{B}\right)$
- 最適決定性オンラインアルゴリズムの期待コスト $= B$

（演習：最適コストの期待値と最適決定性オンラインアルゴリズムの期待コストをきちんと計算せよ）

Yao の原理より以下が成立

定理

スキーレンタル問題に対して，
任意の乱択オンラインアルゴリズムの競合比は $\frac{1}{1-(1-1/B)^B}$ 以上

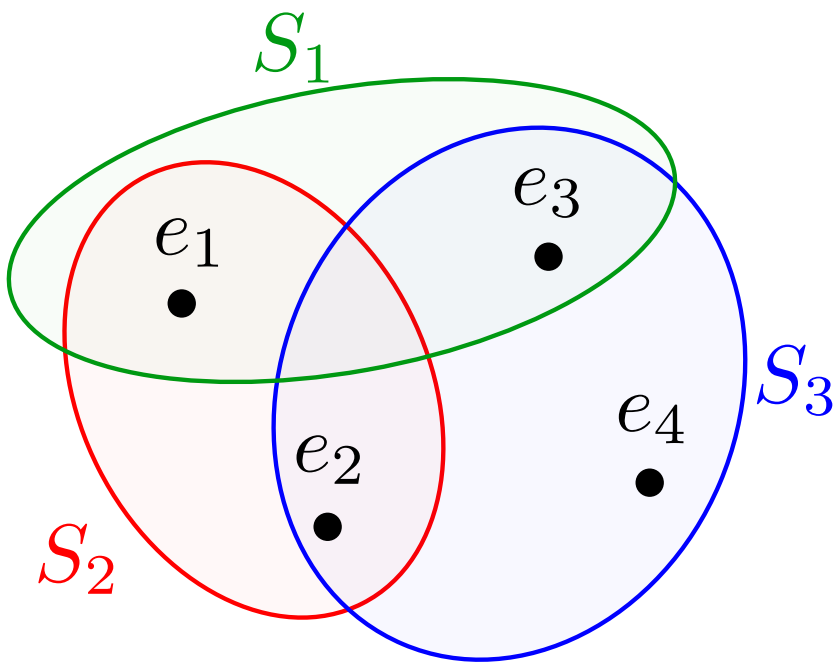
アウトライン

（オフライン）集合被覆問題

入力

- $E = \{e_1, e_2, \ldots, e_m\}$
- $\mathcal{S} = \{S_1, S_2, \ldots, S_n\} \subseteq 2^E$
- $c : \mathcal{S} \to \mathbb{Z}_{++}$

出力 $\min \sum_{S \in \mathcal{S}'} c(S)$ s.t. $\mathcal{S}' \subseteq \mathcal{S}$, $E = \bigcup_{S \in \mathcal{S}'} S$



（オフライン）集合被覆問題

入力

- $E = \{e_1, e_2, \ldots, e_m\}$
- $\mathcal{S} = \{S_1, S_2, \ldots, S_n\} \subseteq 2^E$
- $c : \mathcal{S} \to \mathbb{Z}_{++}$

出力 $\min \sum_{S \in \mathcal{S}'} c(S)$ s.t. $\mathcal{S}' \subseteq \mathcal{S}$, $E = \bigcup_{S \in \mathcal{S}'} S$

主問題

$$
\begin{aligned}
\min \quad & \textstyle\sum_{S \in \mathcal{S}} c(S) \cdot x(S) & \\
\text{s.t.} \quad & \textstyle\sum_{S:\, e \in S} x(S) \geq 1 & (\forall e \in E) \\
& x(S) \in \{0, 1\} & (\forall S \in \mathcal{S})
\end{aligned}
$$

$x(S) \geq 0$ に緩和

双対問題

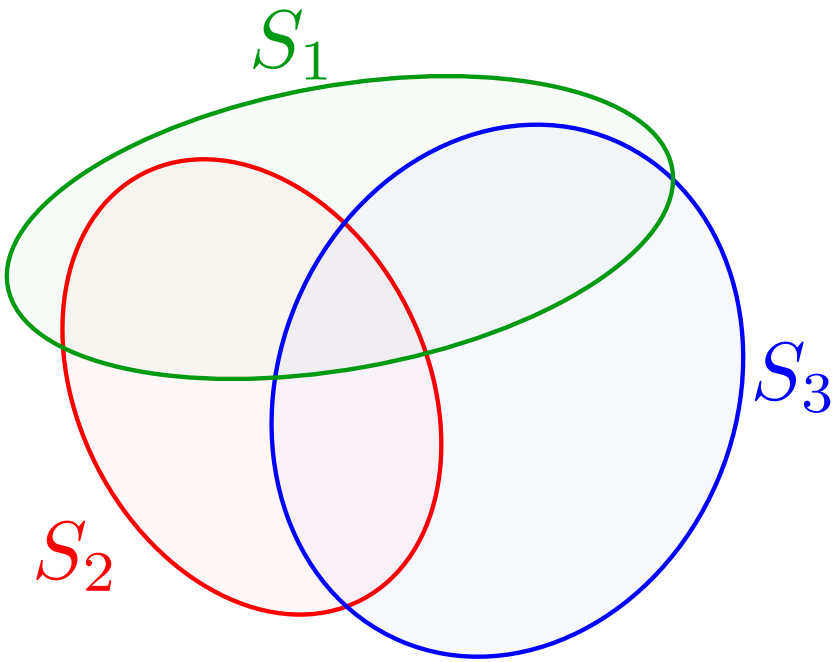
$$
\begin{aligned}
\max \quad & \textstyle\sum_{e \in E} y(e) & \\
\text{s.t.} \quad & \textstyle\sum_{e \in S} y(e) \leq c(S) & (\forall S \in \mathcal{S}) \\
& y(e) \geq 0 & (\forall e \in E)
\end{aligned}
$$

オンライン集合被覆問題

入力

- $E = \{e_1, e_2, \ldots, e_m\}$ (逐次的に与えられる，要素数などは不明)
- $\mathcal{S} = \{S_1, S_2, \ldots, S_n\} \subseteq 2^E$
- $c : \mathcal{S} \to \mathbb{R}_+$

出力 $\min \sum_{S \in \mathcal{S}'} c(S)$ s.t. $\mathcal{S}' \subseteq \mathcal{S}$, $E = \bigcup_{S \in \mathcal{S}'} S$

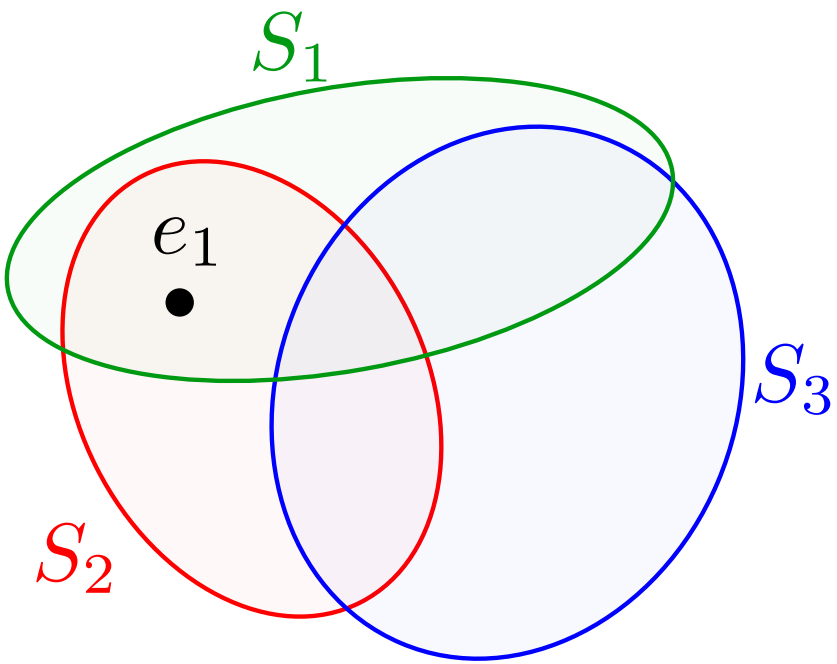


オンライン集合被覆問題

入力

- $E = \{e_1, e_2, \ldots, e_m\}$ (逐次的に与えられる，要素数などは不明)
- $\mathcal{S} = \{S_1, S_2, \ldots, S_n\} \subseteq 2^E$
- $c : \mathcal{S} \to \mathbb{R}_+$

出力 $\min \sum_{S \in \mathcal{S}'} c(S)$ s.t. $\mathcal{S}' \subseteq \mathcal{S}$, $E = \bigcup_{S \in \mathcal{S}'} S$

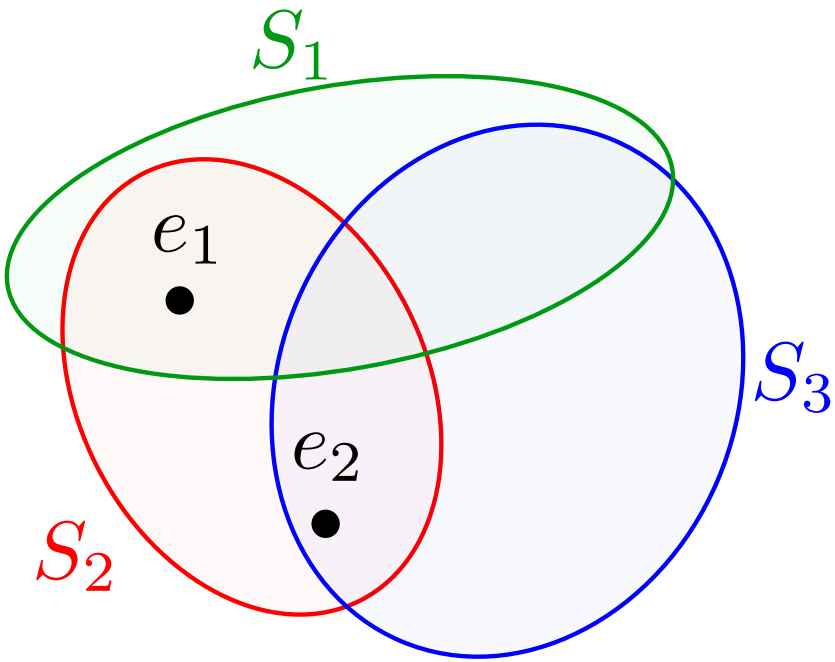


オンライン集合被覆問題

入力

- $E = \{e_1, e_2, \ldots, e_m\}$ (逐次的に与えられる，要素数などは不明)
- $\mathcal{S} = \{S_1, S_2, \ldots, S_n\} \subseteq 2^E$
- $c : \mathcal{S} \to \mathbb{R}_+$

出力 $\min \sum_{S \in \mathcal{S}'} c(S)$ s.t. $\mathcal{S}' \subseteq \mathcal{S}$, $E = \bigcup_{S \in \mathcal{S}'} S$

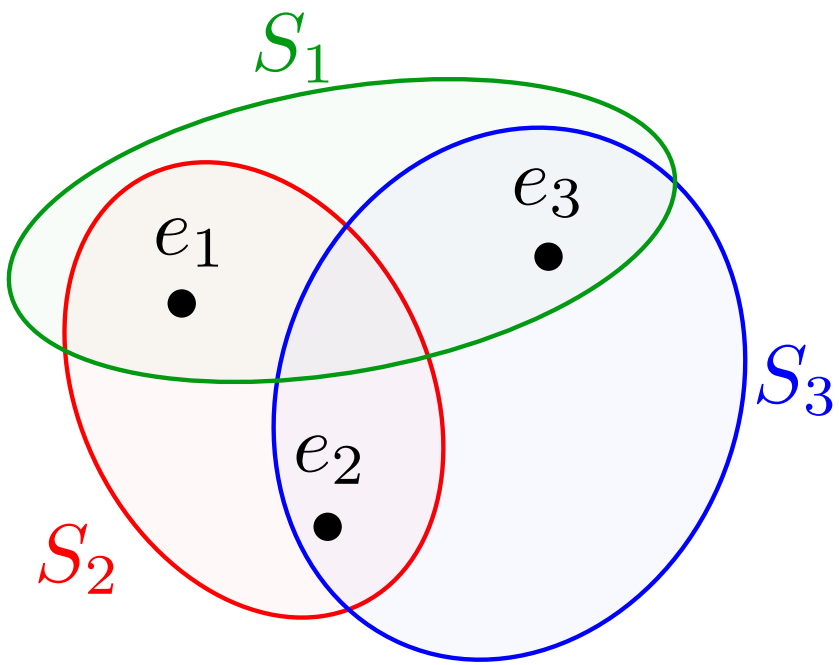


オンライン集合被覆問題

入力

- $E = \{e_1, e_2, \ldots, e_m\}$ (逐次的に与えられる，要素数などは不明)
- $\mathcal{S} = \{S_1, S_2, \ldots, S_n\} \subseteq 2^E$
- $c : \mathcal{S} \to \mathbb{R}_+$

出力 $\min \sum_{S \in \mathcal{S}'} c(S)$ s.t. $\mathcal{S}' \subseteq \mathcal{S}$, $E = \bigcup_{S \in \mathcal{S}'} S$

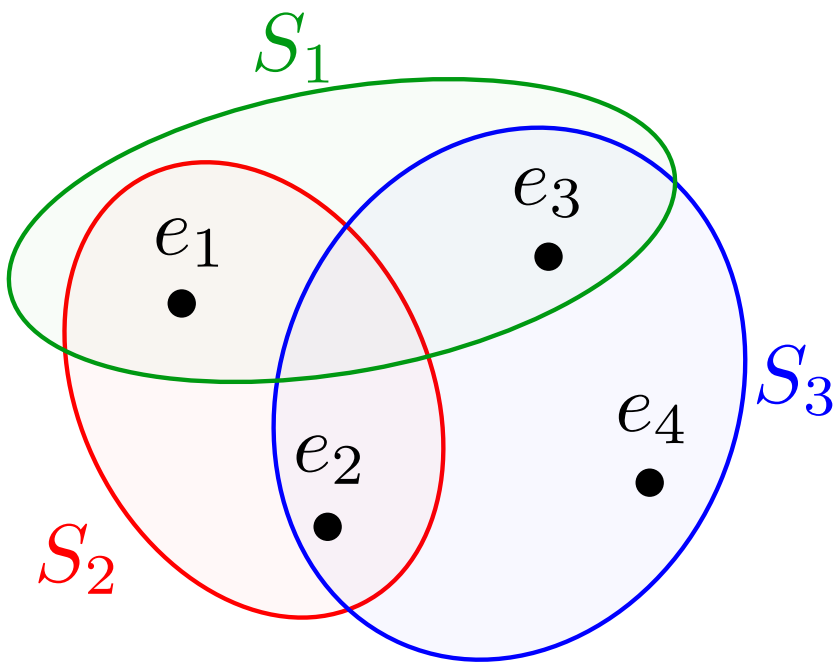


オンライン集合被覆問題

入力

- $E = \{e_1, e_2, \ldots, e_m\}$ (逐次的に与えられる，要素数などは不明)
- $\mathcal{S} = \{S_1, S_2, \ldots, S_n\} \subseteq 2^E$
- $c : \mathcal{S} \to \mathbb{R}_+$

出力 $\min \sum_{S \in \mathcal{S}'} c(S)$ s.t. $\mathcal{S}' \subseteq \mathcal{S}$, $E = \bigcup_{S \in \mathcal{S}'} S$



線形計画による表現

オンライン（小数）集合被覆問題 整数ラウンディングは後で考える

主問題

$$
\begin{aligned}
\min \quad & \textstyle\sum_{S \in \mathcal{S}} c(S) \cdot x(S) & \\
\text{s.t.} \quad & \textstyle\sum_{S:\, e \in S} x(S) \geq 1 & (\forall e \in E) \\
& x(S) \geq 0 & (\forall S \in \mathcal{S})
\end{aligned}
$$

双対問題

$$
\begin{aligned}
\max \quad & \textstyle\sum_{e \in E} y(e) & \\
\text{s.t.} \quad & \textstyle\sum_{e \in S} y(e) \leq c(S) & (\forall S \in \mathcal{S}) \\
& y(e) \geq 0 & (\forall e \in E)
\end{aligned}
$$

- 要素に関する制約が 1 つずつ与えられる
- どのラウンドでも制約を満たしていなければならない
- 変数は増加させることしかできない（一度選択した集合は捨てられない）

アルゴリズム [Buchbinder and Naor 2005]

主問題

$$\begin{aligned} \min \quad & \textstyle\sum_{S \in \mathcal{S}} c(S) \cdot x(S) & \\ \text{s.t.} \quad & \textstyle\sum_{S:\, e \in S} x(S) \geq 1 & (\forall e \in E) \\ & x(S) \geq 0 & (\forall S \in \mathcal{S}) \end{aligned}$$

双対問題

$$\begin{aligned} \max \quad & \textstyle\sum_{e \in E} y(e) & \\ \text{s.t.} \quad & \textstyle\sum_{e \in S} y(e) \leq c(S) & (\forall S \in \mathcal{S}) \\ & y(e) \geq 0 & (\forall e \in E) \end{aligned}$$

アルゴリズム

初期化: $x(S) \leftarrow 0$ $(\forall S \in \mathcal{S})$;

Foreach 要素 e:

$y(e) \leftarrow 0$;

while $\sum_{S:\, e \in S} x(S) < 1$:

(a) $y(e) \leftarrow y(e) + 1$;

(b) $x(S) \leftarrow x(S)(1 + 1/c(S)) + 1/(|\mathcal{S}| \cdot c(S))$ $(\forall S \text{ s.t. } e \in S)$;

アルゴリズムの解析

次の 3 つが成り立つ

1. 主問題について常に実行可能
2. $\Delta P \leq 2\Delta D$
3. 双対問題について実行可能に近い

初期化: $x(S) \leftarrow 0$ $(\forall S \in \mathcal{S})$;

Foreach 要素 e:

$y(e) \leftarrow 0$;

while $\sum_{S:\, e \in S} x(S) < 1$:

(a) $y(e) \leftarrow y(e) + 1$;

(b) $x(S) \leftarrow x(S)(1 + \frac{1}{c(S)}) + \frac{1}{|\mathcal{S}| \cdot c(S)}$ $(\forall S$ s.t. $e \in S)$;

アルゴリズムの解析

次の 3 つが成り立つ

1. 主問題について常に実行可能
2. $\Delta P \leq 2\Delta D$
3. 双対問題について実行可能に近い

初期化: $x(S) \leftarrow 0$ $(\forall S \in \mathcal{S})$;

Foreach 要素 e:

$y(e) \leftarrow 0$;

while $\sum_{S:\, e \in S} x(S) < 1$:

(a) $y(e) \leftarrow y(e) + 1$;

(b) $x(S) \leftarrow x(S)(1 + \frac{1}{c(S)}) + \frac{1}{|\mathcal{S}| \cdot c(S)}$ $(\forall S$ s.t. $e \in S)$;

1. の成立

制約を満たすまで変数を増加

→ 実行可能

主問題

$$
\begin{array}{lll}
\min & \sum_{S \in \mathcal{S}} c(S) \cdot x(S) & \\
\text{s.t.} & \sum_{S:\, e \in S} x(S) \geq 1 & (\forall e \in E) \\
& x(S) \geq 0 & (\forall S \in \mathcal{S})
\end{array}
$$

アルゴリズムの解析

次の 3 つが成り立つ

1. 主問題について常に実行可能
2. $\Delta P \leq 2\Delta D$
3. 双対問題について実行可能に近い

初期化: $x(S) \leftarrow 0$ $(\forall S \in \mathcal{S})$;

Foreach 要素 e:

$y(e) \leftarrow 0$;

while $\sum_{S:\, e \in S} x(S) < 1$:

(a) $y(e) \leftarrow y(e) + 1$;

(b) $x(S) \leftarrow x(S)(1 + \frac{1}{c(S)}) + \frac{1}{|\mathcal{S}| \cdot c(S)}$ $(\forall S$ s.t. $e \in S)$;

2. の成立: 各 $y(e)$ の更新での目的関数値変化を評価

$$\Delta D = \Delta \sum_{e' \in E} y(e') = 1$$

更新前の値

$$\Delta P = \Delta \sum_{S:\, e \in S} x(S) = \sum_{S:\, e \in S} c(S) \left(\frac{x(S)}{c(S)} + \frac{1}{|\mathcal{S}| \cdot c(S)} \right)$$

$$= \sum_{S:\, e \in S} x(S) + \sum_{S:\, e \in S} \frac{1}{|\mathcal{S}|} \leq 2$$

アルゴリズムの解析

次の 3 つが成り立つ

1. 主問題について常に実行可能
2. $\Delta P \leq 2\Delta D$
3. 双対問題について実行可能に近い

初期化: $x(S) \leftarrow 0$ $(\forall S \in \mathcal{S})$;

Foreach 要素 e:

$y(e) \leftarrow 0$;

while $\sum_{S:\, e \in S} x(S) < 1$:

(a) $y(e) \leftarrow y(e) + 1$;

(b) $x(S) \leftarrow x(S)(1 + \frac{1}{c(S)}) + \frac{1}{|\mathcal{S}| \cdot c(S)}$ $(\forall S$ s.t. $e \in S)$;

3. の成立

- $x(S) = \frac{1}{|\mathcal{S}|}\left(\left(1 + \frac{1}{c(S)}\right)^{\sum_{e \in S} y(e)} - 1\right)$

  $\because$ 各更新で $x(S) + 1/|\mathcal{S}|$ は $(1 + 1/c(S))$ 倍される

- $x(S) > 1$ なら $y(e)$ は更新されない

  $\longrightarrow \sum_{e \in S} y(e) = c(S) \cdot O(\log |\mathcal{S}|)$

  $\because (1 + 1/c(S))^{c(S) \cdot \log |\mathcal{S}|} \leq e^{\log |\mathcal{S}|} = |\mathcal{S}|$

双対問題

$$
\begin{array}{lll}
\max & \sum_{e \in E} y(e) & \\
\text{s.t.} & \sum_{e \in S} y(e) \leq c(S) & (\forall S \in \mathcal{S}) \\
& y(e) \geq 0 & (\forall e \in E)
\end{array}
$$

アルゴリズムの解析

次の 3 つが成り立つ

1. 主問題について常に実行可能
2. $\Delta P \leq 2\Delta D$
3. 双対問題について実行可能に近い

初期化: $x(S) \leftarrow 0$ $(\forall S \in \mathcal{S})$;

Foreach 要素 e:

$y(e) \leftarrow 0$;

while $\sum_{S:\, e \in S} x(S) < 1$:

(a) $y(e) \leftarrow y(e) + 1$;

(b) $x(S) \leftarrow x(S)(1 + \frac{1}{c(S)}) + \frac{1}{|\mathcal{S}| \cdot c(S)}$ $(\forall S \text{ s.t. } e \in S)$;

まとめ

- 双対問題の解は $O(\log |\mathcal{S}|)$ で割れば実行可能
- アルゴリズムの競合比は $O(\log |\mathcal{S}|)$
- $d := \max_{e \in E} \{S \in \mathcal{S} \mid e \in S\}$ が既知ならば競合比 $O(\log d)$ にできる
- 双対問題（詰込問題）についても，各ラウンドで与えられた変数について変更するような，競合比 2 で各制約を高々 $O(\log |\mathcal{S}|)$ 倍しか破らない解を出力するアルゴリズムだと解釈可能

下界 [Buchbinder and Naor 2005]

$n := |\mathcal{S}|$. $\sigma : [n] \to [n]$ をランダムなパーミュテーションとする

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} x_i \\
\text{s.t.} \quad & x_i \geq 0 \quad (\forall i \in [n]) \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} + x_{\sigma(n)} \geq 1 \qquad \mathbb{E}[x_{\sigma(n)}] \geq \tfrac{1}{n} \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} \geq 1 \qquad \mathbb{E}[x_{\sigma(n-1)}] \geq \tfrac{1}{n-1} \\
& \vdots \qquad\qquad\qquad\qquad\qquad\quad \vdots \\
& x_{\sigma(1)} \geq 1 \qquad\qquad\qquad\qquad\quad \mathbb{E}[x_{\sigma(1)}] \geq 1
\end{aligned}
$$

下界 [Buchbinder and Naor 2005]

$n := |\mathcal{S}|$. $\sigma : [n] \to [n]$ をランダムなパーミュテーションとする

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} x_i \\ \text{s.t.} \quad & x_i \geq 0 \quad (\forall i \in [n]) \\ & x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} + x_{\sigma(n)} \geq 1 \qquad \mathbb{E}[x_{\sigma(n)}] \geq \tfrac{1}{n} \\ & x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} \geq 1 \qquad \mathbb{E}[x_{\sigma(n-1)}] \geq \tfrac{1}{n-1} \\ & \vdots \qquad\qquad\qquad\qquad \vdots \\ & x_{\sigma(1)} \geq 1 \qquad\qquad\qquad\qquad \mathbb{E}[x_{\sigma(1)}] \geq 1 \end{aligned}$$

下界 [Buchbinder and Naor 2005]

$n := |\mathcal{S}|$. $\sigma : [n] \to [n]$ をランダムなパーミュテーションとする

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} x_i \\
\text{s.t.} \quad & x_i \geq 0 \quad (\forall i \in [n]) \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} + x_{\sigma(n)} \geq 1 && \mathbb{E}[x_{\sigma(n)}] \geq \tfrac{1}{n} \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} \geq 1 && \mathbb{E}[x_{\sigma(n-1)}] \geq \tfrac{1}{n-1} \\
& \vdots && \vdots \\
& x_{\sigma(1)} \geq 1 && \mathbb{E}[x_{\sigma(1)}] \geq 1
\end{aligned}
$$

下界 [Buchbinder and Naor 2005]

$n := |\mathcal{S}|$. $\sigma : [n] \to [n]$ をランダムなパーミュテーションとする

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} x_i \\
\text{s.t.} \quad & x_i \geq 0 \quad (\forall i \in [n]) \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} + x_{\sigma(n)} \geq 1 && \mathbb{E}[x_{\sigma(n)}] \geq \tfrac{1}{n} \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} \geq 1 && \mathbb{E}[x_{\sigma(n-1)}] \geq \tfrac{1}{n-1} \\
& \vdots && \vdots \\
& x_{\sigma(1)} \geq 1 && \mathbb{E}[x_{\sigma(1)}] \geq 1
\end{aligned}
$$

下界 [Buchbinder and Naor 2005]

$n := |\mathcal{S}|$. $\sigma : [n] \to [n]$ をランダムなパーミュテーションとする

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} x_i \\
\text{s.t.} \quad & x_i \geq 0 \quad (\forall i \in [n]) \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} + x_{\sigma(n)} \geq 1 && \mathbb{E}[x_{\sigma(n)}] \geq \tfrac{1}{n} \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} \geq 1 && \mathbb{E}[x_{\sigma(n-1)}] \geq \tfrac{1}{n-1} \\
& \vdots && \vdots \\
& x_{\sigma(1)} \geq 1 && \mathbb{E}[x_{\sigma(1)}] \geq 1
\end{aligned}
$$

下界 [Buchbinder and Naor 2005]

$n := |\mathcal{S}|$. $\sigma : [n] \to [n]$ をランダムなパーミュテーションとする

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} x_i \\
\text{s.t.} \quad & x_i \geq 0 \quad (\forall i \in [n]) \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} + x_{\sigma(n)} \geq 1 && \mathbb{E}[x_{\sigma(n)}] \geq \tfrac{1}{n} \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} \geq 1 && \mathbb{E}[x_{\sigma(n-1)}] \geq \tfrac{1}{n-1} \\
& \vdots && \vdots \\
& x_{\sigma(1)} \geq 1 && \mathbb{E}[x_{\sigma(1)}] \geq 1
\end{aligned}
$$

下界 [Buchbinder and Naor 2005]

$n := |\mathcal{S}|$. $\sigma : [n] \to [n]$ をランダムなパーミュテーションとする

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} x_i \\
\text{s.t.} \quad & x_i \geq 0 \quad (\forall i \in [n]) \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} + x_{\sigma(n)} \geq 1 && \mathbb{E}[x_{\sigma(n)}] \geq \tfrac{1}{n} \\
& x_{\sigma(1)} + \cdots + x_{\sigma(n-1)} \geq 1 && \mathbb{E}[x_{\sigma(n-1)}] \geq \tfrac{1}{n-1} \\
& \vdots && \vdots \\
& x_{\sigma(1)} \geq 1 && \mathbb{E}[x_{\sigma(1)}] \geq 1
\end{aligned}
$$

- $\forall$ アルゴリズムの期待コスト $\geq \frac{1}{n} + \frac{1}{n-1} + \cdots + 1 = \Omega(\log n)$
- 最適コストは 1 ($x_{\sigma(1)} = 1$)

→ Yao の原理より任意の乱択アルゴリズムの競合比は $\Omega(\log n)$

乱択丸め

元問題

$$\begin{aligned} \min \quad & \textstyle\sum_{S \in \mathcal{S}} c(S) \cdot x(S) & \\ \text{s.t.} \quad & \textstyle\sum_{S:\, e \in S} x(S) \geq 1 & (\forall e \in E) \\ & x(S) \in \{0, 1\} & (\forall S \in \mathcal{S}) \end{aligned}$$

緩和問題

$$\begin{aligned} \min \quad & \textstyle\sum_{S \in \mathcal{S}} c(S) \cdot x(S) & \\ \text{s.t.} \quad & \textstyle\sum_{S:\, e \in S} x(S) \geq 1 & (\forall e \in E) \\ & x(S) \geq 0 & (\forall S \in \mathcal{S}) \end{aligned}$$

$[0, 1]$ 上の一様乱数 $d(S)$ をあらかじめ選び $x(S) \geq d(S)$ となれば選択

- 単純に確率 $x(S)$ で S を選択だと高確率で実行不能
  - 各制約を満たす確率は $1 - \prod_{S:\, e \in S}(1 - x(S)) \geq 1 - \frac{1}{e}$
  - ただし制約は $|E|$ 個あるので，全て満たす確率は低い
- $d(S)$ が $[0, 1]$ 上の一様乱数 $O(\log |E|)$ 個の最小値 $\Rightarrow$ 高確率で実行可能

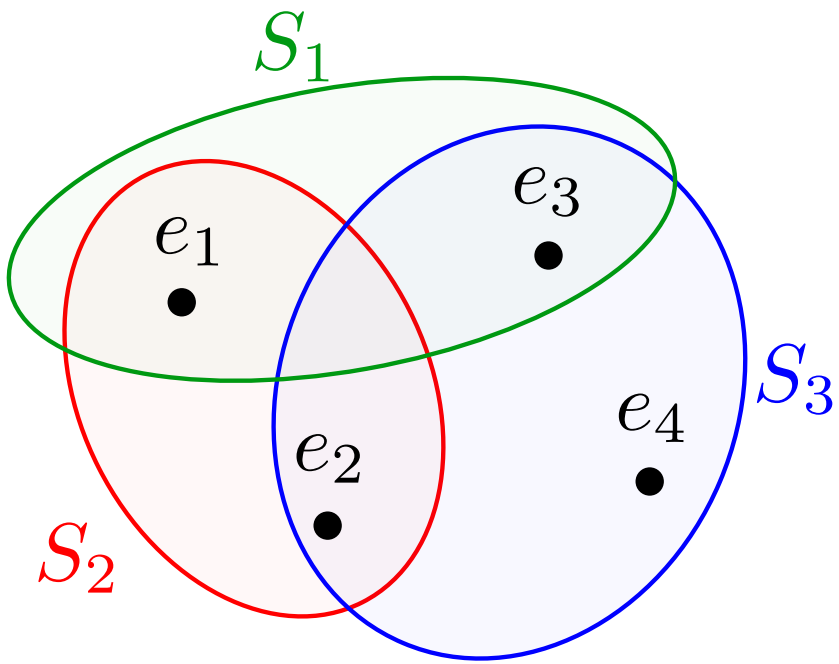
  $|E|$ は不明だが，逐次的に乱数の個数を増やせばよい
- 競合比は $O(\log |\mathcal{S}| \cdot \log |E|)$

脱乱択化

- 任意の決定性アルゴリズムの競合比は $\Omega(|E|)$

  $S_j = \{e_1, \ldots, e_j\}$, $c(S_j) = 1$ を考える

- E と $\mathcal{S}$ が既知であるが，与えられるのは未知の $E' \subseteq E$ であり，
  与えられる順番が未知の場合は競合比 $O(\log |\mathcal{S}| \cdot \log |E|)$ を達成可能



アウトライン

(小数) 被覆問題

集合被覆問題

$$
\begin{aligned}
\min \quad & \textstyle\sum_{S \in \mathcal{S}} c(S) \cdot x(S) & \\
\text{s.t.} \quad & \textstyle\sum_{S:\, e \in S} x(S) \geq 1 & (\forall e \in E) \\
& x(S) \geq 0 & (\forall S \in \mathcal{S})
\end{aligned}
$$

被覆問題

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} c_i x_i & \\
\text{s.t.} \quad & \textstyle\sum_{i=1}^{n} a_{i,j} x_i \geq b_j & (\forall j \in [m]) \\
& x_i \geq 0 & (\forall i \in [n])
\end{aligned}
$$

- $a_{i,j},\ b_j,\ c_i$ は非負と仮定
- $b_j = 0$ の時，制約 $\sum_{i=1}^{n} a_{i,j} x_i \geq b_j$ は何も制限しない
  → 削除しても構わない
- $b_j > 0$ の時，制約 $\sum_{i=1}^{n} a_{i,j} x_i \geq b_j$ の両辺を b_j で割ることができる
  → 制約 $\sum_{i=1}^{n} a'_{i,j} x_i \geq 1$ を考えれば良い

オンライン被覆問題

主問題（被覆問題）

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} c_i x_i & \\
\text{s.t.} \quad & \textstyle\sum_{i=1}^{n} a_{i,j} x_i \geq 1 & (\forall j \in [m]) \\
& x_i \geq 0 & (\forall i \in [n])
\end{aligned}
$$

双対問題（詰込問題）

$$
\begin{aligned}
\max \quad & \textstyle\sum_{j=1}^{m} y_j & \\
\text{s.t.} \quad & \textstyle\sum_{j=1}^{m} a_{i,j} y_j \leq c_i & (\forall i \in [n]) \\
& y_j \geq 0 & (\forall j \in [m])
\end{aligned}
$$

- ラウンド j では j 番目の制約が与えられる
- どのラウンドでも与えられた制約を全て満たしていなければならない
- (主問題の) 変数は増加させることしかできない

アルゴリズム

主問題

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} c_i x_i & \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{n} a_{i,j} x_i \geq 1 & (\forall j \in [m]) \\ & x_i \geq 0 & (\forall i \in [n]) \end{aligned}$$

双対問題

$$\begin{aligned} \max \quad & \textstyle\sum_{j=1}^{m} y_j & \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{m} a_{i,j} y_j \leq c_i & (\forall i \in [n]) \\ & y_j \geq 0 & (\forall j \in [m]) \end{aligned}$$

アルゴリズムの準備

- 最適値をダブリングにより推定
- フェーズ 1 では $\alpha_1 \leftarrow \min_{i=1}^{n} \frac{c_i}{a_{i,1}}$ に設定
- アルゴリズムの主問題解の目的関数値が α_r を超えたら，
  $\alpha_{r+1} \leftarrow 2\alpha_r$ として次のフェーズを開始
- 各フェーズでは変数をリセットし，0 から増加し直す

  実際には減らすことはできないので，$\max_r x_i^{(r)}$ を真の変数の値として実行する

アルゴリズム

主問題

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} c_i x_i & \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_{i,j} x_i \geq 1 & (\forall j \in [m]) \\ & x_i \geq 0 & (\forall i \in [n]) \end{aligned}$$

双対問題

$$\begin{aligned} \max \quad & \sum_{j=1}^{m} y_j & \\ \text{s.t.} \quad & \sum_{j=1}^{m} a_{i,j} y_j \leq c_i & (\forall i \in [n]) \\ & y_j \geq 0 & (\forall j \in [m]) \end{aligned}$$

アルゴリズム (フェーズ r)

初期化: $x_i \leftarrow 0$ $(\forall i \in [n])$;

Foreach j:

$y_j \leftarrow 0$;

while $\sum_{i=1}^{n} a_{i,j} x_i < 1$:

(a) y_j を連続的に増加;

(b) $x_i \leftarrow \frac{\alpha_r}{2nc_i} \cdot \exp\left(\frac{\log 2n}{c_i} \sum_{k=1}^{j} a_{i,k} y_k\right)$ $(\forall i \in [n])$;

アルゴリズムの解析

次の 4 つが成り立つ

1. 主問題について実行可能
2. 双対問題について実行可能
3. フェーズ r 終了時，$P_r \leq 2\alpha_r$
4. フェーズ r 終了時，$D_r \geq \frac{\alpha_r}{2 \log 2n}$

初期化: $x_i \leftarrow 0$ $(\forall i \in [n])$;

Foreach j:

$y_j \leftarrow 0$;
while $\sum_{i=1}^{n} a_{i,j} x_i < 1$:

(a) y_j を連続的に増加;

(b) $x_i \leftarrow \frac{\alpha_r}{2nc_i} \cdot \exp\left(\frac{\log 2n}{c_i} \sum_{k=1}^{j} a_{i,k} y_k\right)$ $(\forall i \in [n])$;

アルゴリズムの解析

次の 4 つが成り立つ

1. 主問題について実行可能
2. 双対問題について実行可能
3. フェーズ r 終了時，$P_r \leq 2\alpha_r$
4. フェーズ r 終了時，$D_r \geq \frac{\alpha_r}{2 \log 2n}$

初期化: $x_i \leftarrow 0$ $(\forall i \in [n])$;

Foreach j:

$y_j \leftarrow 0$;
while $\sum_{i=1}^{n} a_{i,j} x_i < 1$:

(a) y_j を連続的に増加;

(b) $x_i \leftarrow \frac{\alpha_r}{2nc_i} \cdot \exp\left(\frac{\log 2n}{c_i} \sum_{k=1}^{j} a_{i,k} y_k\right)$ $(\forall i \in [n])$;

1. の成立

- 各 x_i は単調非減少
- 各ラウンドでは新しい制約を満たすまで変数 x_i を増加 ⟶ 実行可能

主問題

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} c_i x_i & \\
\text{s.t.} \quad & \sum_{i=1}^{n} a_{i,j} x_i \geq 1 & (\forall j \in [m]) \\
& x_i \geq 0 & (\forall i \in [n])
\end{aligned}
$$

アルゴリズムの解析

次の 4 つが成り立つ

1. 主問題について実行可能
2. 双対問題について実行可能
3. フェーズ r 終了時，$P_r \leq 2\alpha_r$
4. フェーズ r 終了時，$D_r \geq \frac{\alpha_r}{2 \log 2n}$

初期化: $x_i \leftarrow 0$ $(\forall i \in [n])$;

Foreach j:

$y_j \leftarrow 0$;
while $\sum_{i=1}^{n} a_{i,j} x_i < 1$:

(a) y_j を連続的に増加;

(b) $x_i \leftarrow \frac{\alpha_r}{2nc_i} \cdot \exp\left(\frac{\log 2n}{c_i} \sum_{k=1}^{j} a_{i,k} y_k\right)$ $(\forall i \in [n])$;

2. の成立

フェーズ r において

- $x_i \leq \alpha_r / c_i$ (フェーズ r では主問題の目的関数値が α_r 以下)
- $x_i = \frac{\alpha_r}{2nc_i} \cdot \exp\left(\frac{\log 2n}{c_i} \sum_{k=1}^{j} a_{i,k} y_k\right)$

→ $\sum_{k=1}^{j} a_{i,k} y_k \leq c_i$

アルゴリズムの解析

次の 4 つが成り立つ

1. 主問題について実行可能
2. 双対問題について実行可能
3. フェーズ r 終了時，$P_r \leq 2\alpha_r$
4. フェーズ r 終了時，$D_r \geq \frac{\alpha_r}{2 \log 2n}$

初期化: $x_i \leftarrow 0$ $(\forall i \in [n])$;

Foreach j:

$y_j \leftarrow 0$;
while $\sum_{i=1}^{n} a_{i,j} x_i < 1$:

(a) y_j を連続的に増加;

(b) $x_i \leftarrow \frac{\alpha_r}{2nc_i} \cdot \exp\left(\frac{\log 2n}{c_i} \sum_{k=1}^{j} a_{i,k} y_k\right)$ $(\forall i \in [n])$;

3. の成立

$$P_r \leq \alpha_r + \alpha_{r-1} + \cdots \leq \alpha_r + \frac{1}{2}\alpha_r + \frac{1}{4}\alpha_r + \cdots \leq 2\alpha_r$$

アルゴリズムの解析

次の 4 つが成り立つ

1. 主問題について実行可能
2. 双対問題について実行可能
3. フェーズ r 終了時，$P_r \leq 2\alpha_r$
4. フェーズ r 終了時，$D_r \geq \frac{\alpha_r}{2 \log 2n}$

初期化: $x_i \leftarrow 0$ $(\forall i \in [n])$;

Foreach j:

$y_j \leftarrow 0$;
while $\sum_{i=1}^{n} a_{i,j} x_i < 1$:

(a) y_j を連続的に増加;

(b) $x_i \leftarrow \frac{\alpha_r}{2nc_i} \cdot \exp\left(\frac{\log 2n}{c_i} \sum_{k=1}^{j} a_{i,k} y_k\right)$ $(\forall i \in [n])$;

4. の成立

- フェーズ r の開始直後，$\sum_{i=1}^{n} c_i x_i^{(r)} = \sum_{i=1}^{n} c_i \cdot \frac{\alpha_r}{2nc_i} = \frac{\alpha_r}{2}$
- フェーズ r の終了時，$\sum_{i=1}^{n} c_i x_i^{(r)} = \alpha_r$
- $\frac{\partial \sum_{i=1}^{n} c_i x^{(r)}}{\partial y_j} = \log 2n \cdot \sum_{i=1}^{n} a_{i,k} x_i \leq \log 2n$

→ $D_r \geq (\alpha_r - \frac{\alpha_r}{2}) \cdot \frac{1}{\log 2n} = \frac{\alpha_r}{2 \log 2n}$

アルゴリズムの解析

次の 4 つが成り立つ

1. 主問題について実行可能
2. 双対問題について実行可能
3. フェーズ r 終了時，$P_r \leq 2\alpha_r$
4. フェーズ r 終了時，$D_r \geq \frac{\alpha_r}{2\log 2n}$

初期化: $x_i \leftarrow 0$ $(\forall i \in [n])$;

Foreach j:

$y_j \leftarrow 0$;
while $\sum_{i=1}^{n} a_{i,j} x_i < 1$:

(a) y_j を連続的に増加;

(b) $x_i \leftarrow \frac{\alpha_r}{2nc_i} \cdot \exp\left(\frac{\log 2n}{c_i} \sum_{k=1}^{j} a_{i,k} y_k\right)$ $(\forall i \in [n])$;

まとめ

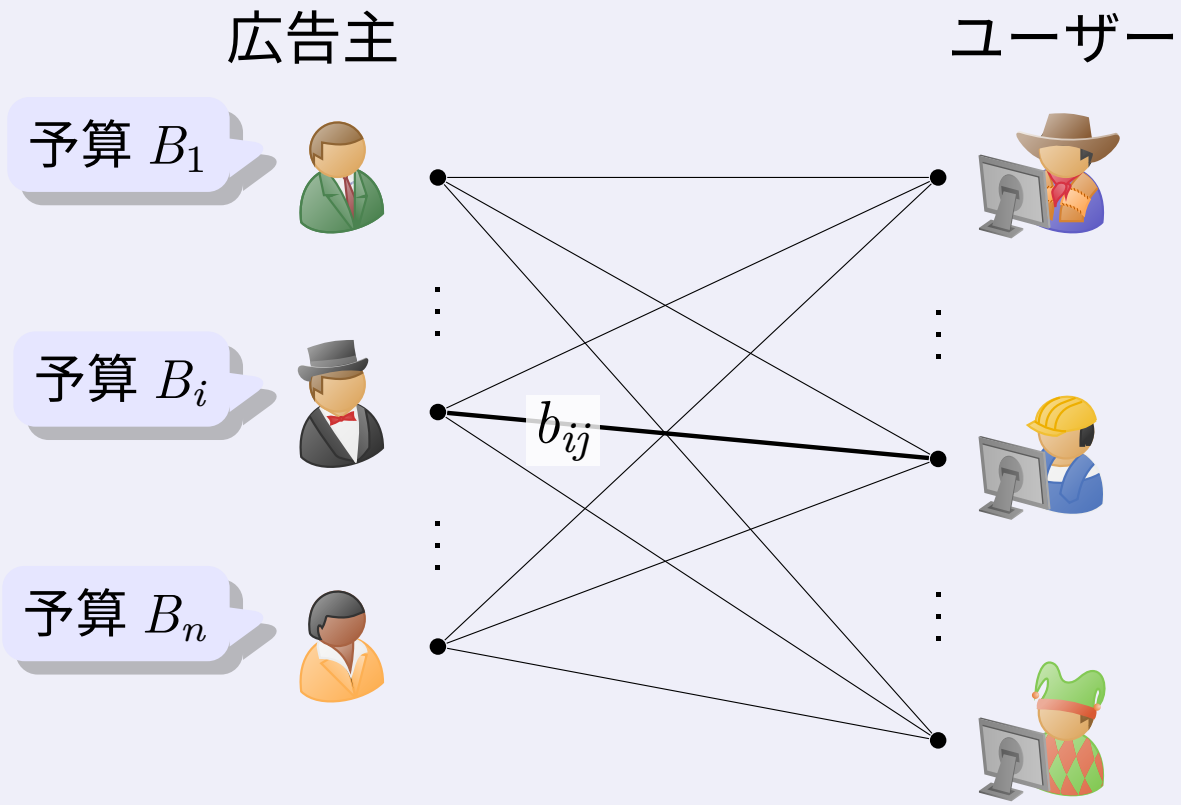
- 最終的なフェーズを r とすると，
  $P \leq 2\alpha_r \leq 4\alpha_{r-1} \leq (8\log 2n) D_{r-1} \leq (8\log 2n)\mathrm{OPT}$
- アルゴリズムの競合比は $O(\log n)$ [Buchbinder and Naor 2005]

アウトライン

インターネット広告

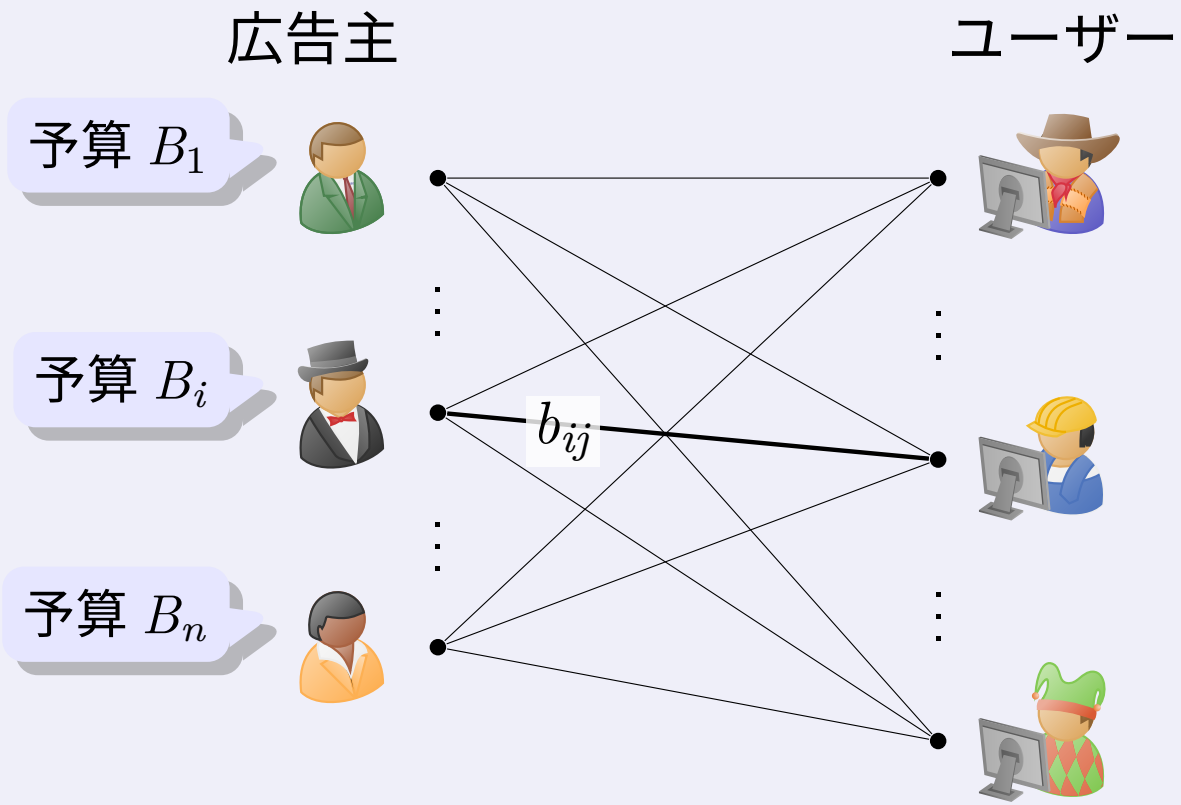
https://search.yahoo.co.jp/

（オフライン）アドワーズ問題



- 広告主 i は，ユーザー j に広告を見せられれば，オーナーに b_{ij} 支払う
- 各ユーザー j には，高々 1 人の広告主を割り当てる
- 各広告主 i は予算 B_i までしか支払うことができない
- 目標はオーナーの収入最大化

（オフライン）アドワーズ問題

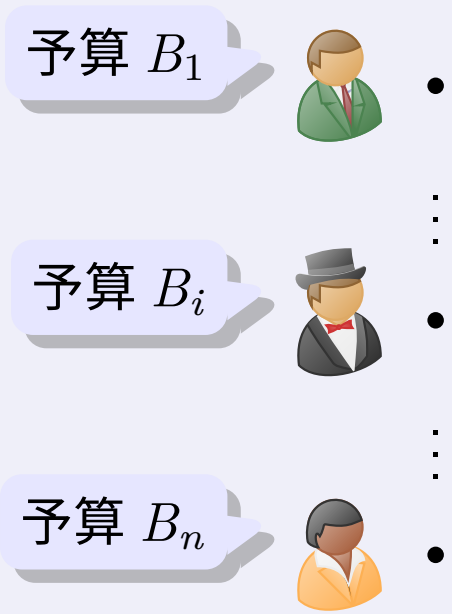


式で書くと

$$
\begin{array}{lll}
\max & \sum_{j \in M} \sum_{i \in N} b_{ij} x_{ij} & \\
\text{s.t.} & \sum_{j \in M} b_{ij} x_{ij} \leq B_i & (\forall i \in N) \\
& \sum_{i \in N} x_{ij} \leq 1 & (\forall j \in M) \\
& x_{ij} \in \{0, 1\} & (\forall i \in N, \forall j \in M)
\end{array}
$$

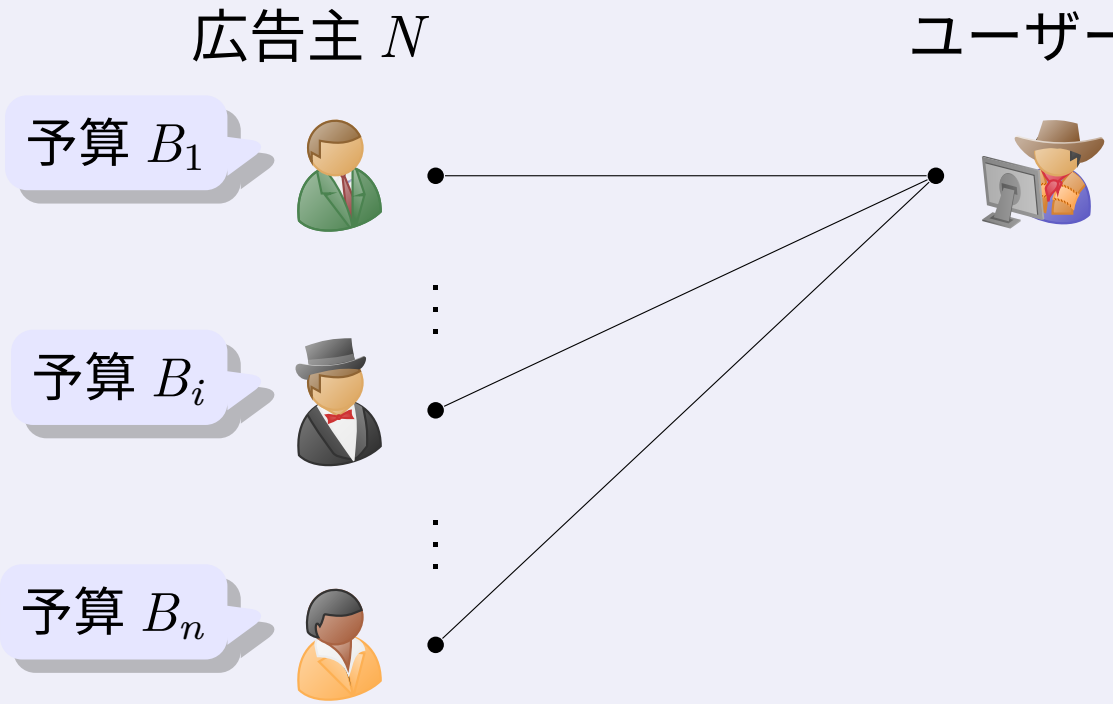
アドワーズ問題 [Mehta et al. 2005]

- ユーザーが逐次的にやってくる
- 来たユーザーに対しその場でどの広告を割り当てるか決める



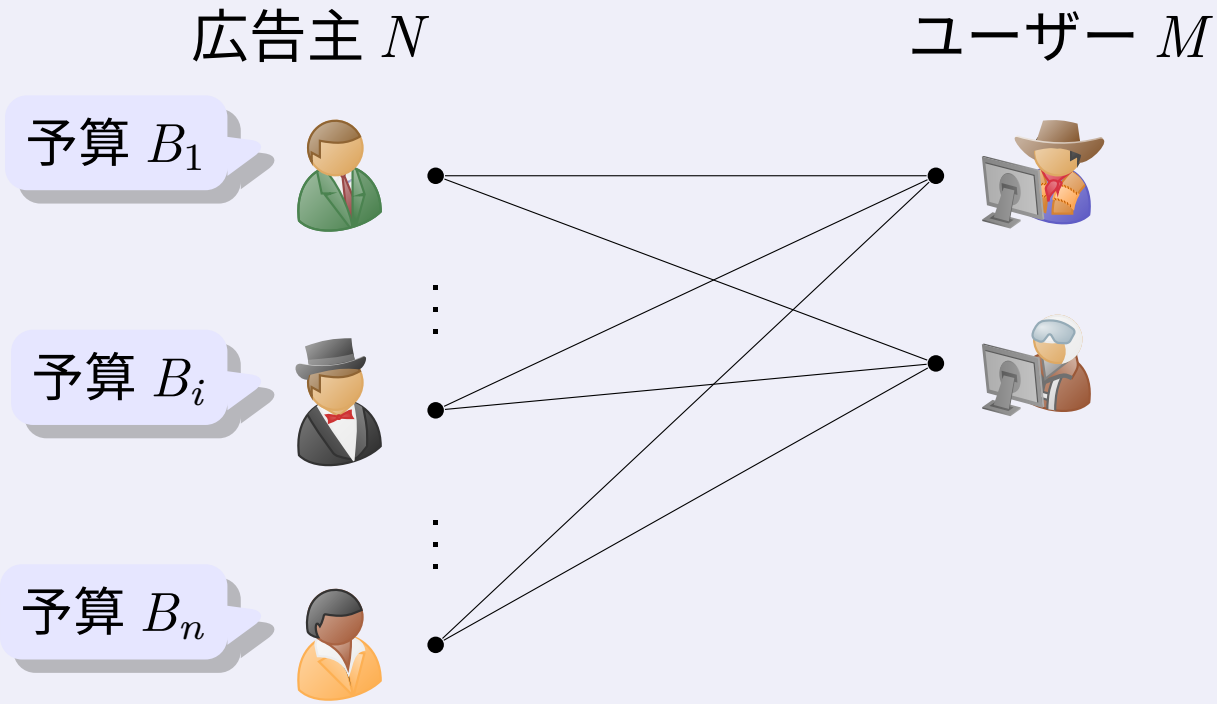
アドワーズ問題 [Mehta et al. 2005]

- ユーザーが逐次的にやってくる
- 来たユーザーに対しその場でどの広告を割り当てるか決める



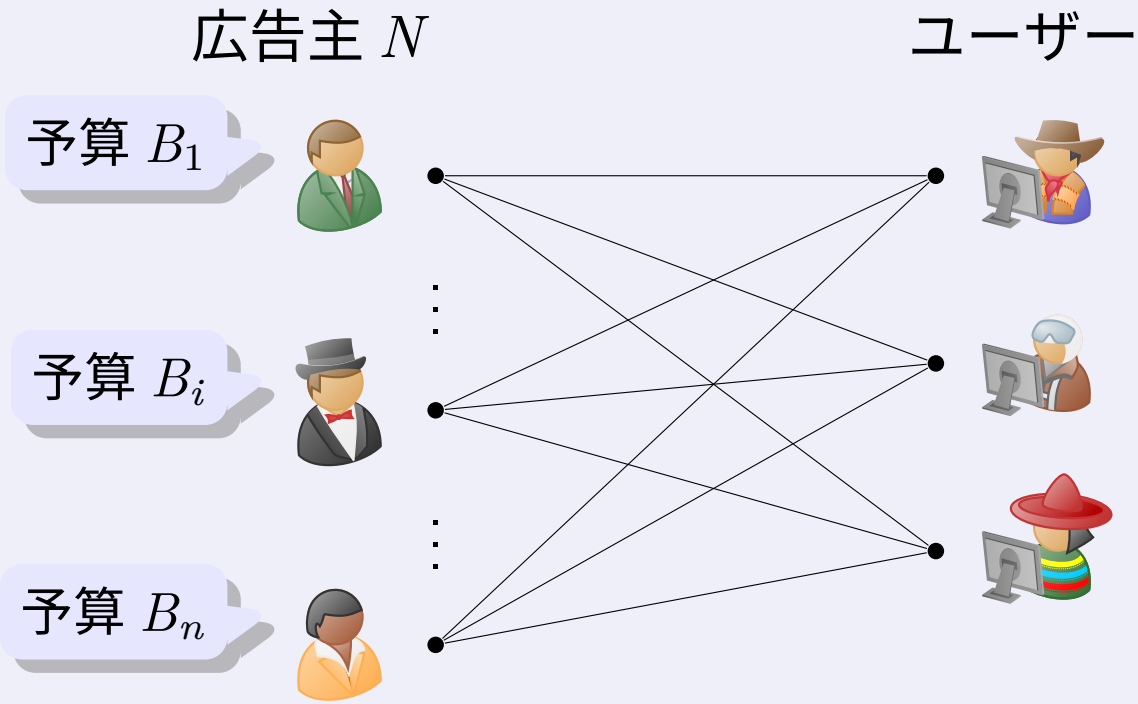
アドワーズ問題 [Mehta et al. 2005]

- ユーザーが逐次的にやってくる
- 来たユーザーに対しその場でどの広告を割り当てるか決める



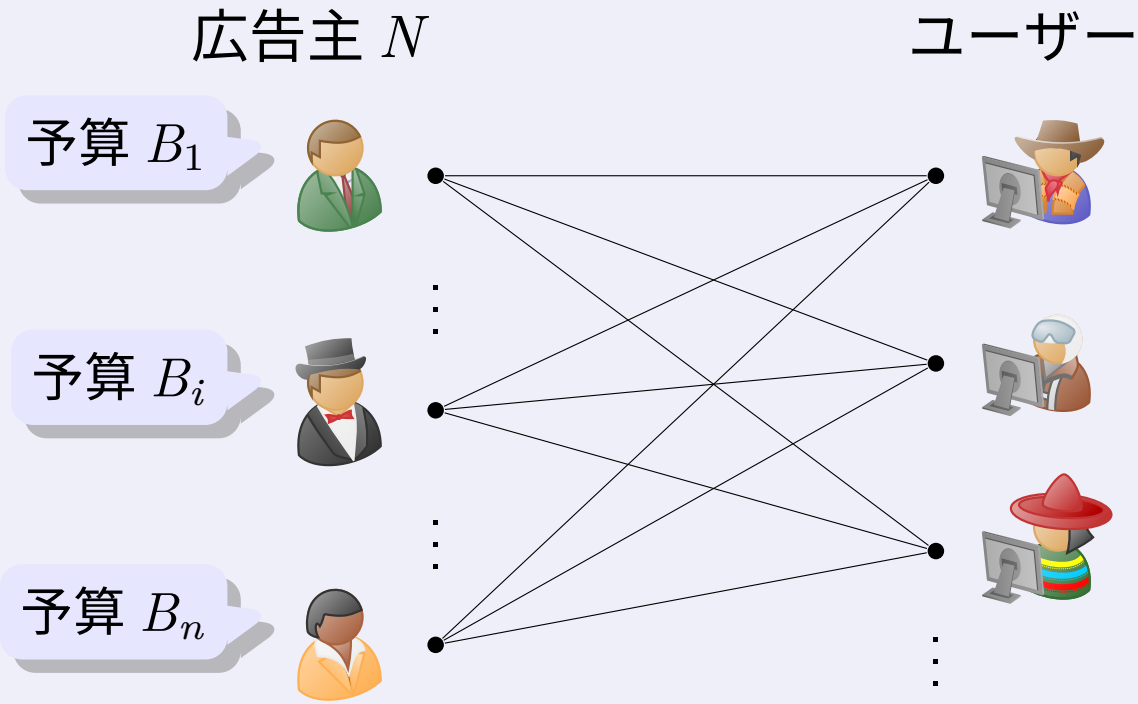
アドワーズ問題 [Mehta et al. 2005]

- ユーザーが逐次的にやってくる
- 来たユーザーに対しその場でどの広告を割り当てるか決める



アドワーズ問題 [Mehta et al. 2005]

- ユーザーが逐次的にやってくる
- 来たユーザーに対しその場でどの広告を割り当てるか決める



Small bid assumption

予算に対して入札額は小さいと仮定

任意の i, j について $b_{ij}/B_i \leq R_{\max}$ $(\ll 1)$

- 1000 回の表示で数十〜数百円程度 ⟶ 実用的に妥当な仮定
- 広告主が 1 人でも競合比は $1 - \Omega(R_{\max})$ 以下 ⟶ 解析的に必要な仮定

線形緩和

主問題

$$
\begin{array}{lll}
\max & \sum_{j\in M}\sum_{i\in N} b_{ij}x_{ij} & \\
\text{s.t.} & \sum_{j\in M} b_{ij}x_{ij} \le B_i & (\forall i \in N) \\
& \sum_{i\in N} x_{ij} \le 1 & (\forall j \in M) \\
& x_{ij} \ge 0 & (\forall i \in N, \forall j \in M)
\end{array}
$$

双対問題

$$
\begin{array}{lll}
\min & \sum_{i\in N} B_i y_i + \sum_{j\in M} z_j & \\
\text{s.t.} & z_j \ge b_{ij}(1-y_i) & (\forall i \in N, \forall j \in M) \\
& \sum_{i\in N} x_{ij} \le 1 & (\forall j \in M) \\
& y_i \ge 0,\ z_j \ge 0 & (\forall i \in N,\ \forall j \in M)
\end{array}
$$

仮定

- b_{ij}, B_i は非負，$b_{ij}/B_i \le R_{\max}$
- 各ラウンドでは新しい j がやってきて，x_{ij} $(\forall i)$ を決定する

アルゴリズム

主問題

$$
\begin{aligned}
\max\ & \textstyle\sum_{j\in M}\sum_{i\in N} b_{ij}x_{ij} & \\
\text{s.t.}\ & \textstyle\sum_{j\in M} b_{ij}x_{ij} \le B_i & (\forall i \in N)\\
& \textstyle\sum_{i\in N} x_{ij} \le 1 & (\forall j \in M)\\
& x_{ij} \ge 0 & (\forall i \in N, \forall j \in M)
\end{aligned}
$$

双対問題

$$
\begin{aligned}
\min\ & \textstyle\sum_{i\in N} B_i y_i + \sum_{j\in M} z_j & \\
\text{s.t.}\ & z_j \ge b_{ij}(1-y_i) & (\forall i \in N, \forall j \in M)\\
& y_i \ge 0,\ z_j \ge 0 & (\forall i \in N,\ \forall j \in M)
\end{aligned}
$$

アルゴリズム

初期化: $y_i \leftarrow 0\ (\forall i \in [n])$;

Foreach j:

$i^* \in \arg\max_{i\in N}\{b_{ij}(1-y_i)\}$ を選択;

if $y_{i^*} < 1$:

$x_{i^*,j} \leftarrow 1$; //i^* に j を (予算がまだ残っているなら) 割当

$z_j \leftarrow \max\{0,\ b_{i^*j}(1-y_{i^*})\}$;

$$y_{i^*} \leftarrow y_{i^*}\cdot\left(1+\frac{b_{i^*j}}{B_{i^*}}\right)+\frac{b_{i^*j}}{(c-1)B_{i^*}};$$

$(1+R_{\max})^{1/R_{\max}}$

アルゴリズムの解析

次の 3 つが成り立つ

1. 双対問題について常に実行可能
2. 主問題についてほぼ実行可能

   各 i について予算超過は高々 1 回
3. $\Delta D \leq (1 + \frac{1}{c-1})\Delta P$

初期化: $y_i \leftarrow 0\ (\forall i \in [n])$;

Foreach j:

$i^* \in \arg\max_{i \in N}\{b_{ij}(1 - y_i)\}$ を選択;

if $y_{i^*} < 1$:

$x_{i^*,j} \leftarrow 1$;

$z_j \leftarrow \max\{0,\ b_{i^*j}(1 - y_{i^*})\}$;

$y_{i^*} \leftarrow y_{i^*} \cdot \left(1 + \frac{b_{i^*j}}{B_{i^*}}\right) + \frac{b_{i^*j}}{(c-1)B_{i^*}}$;

アルゴリズムの解析

次の 3 つが成り立つ

1. 双対問題について常に実行可能
2. 主問題についてほぼ実行可能
   各 i について予算超過は高々 1 回
3. $\Delta D \leq (1 + \frac{1}{c-1})\Delta P$

初期化: $y_i \leftarrow 0\ (\forall i \in [n])$;

Foreach j:

  $i^* \in \arg\max_{i \in N}\{b_{ij}(1 - y_i)\}$ を選択;

  if $y_{i^*} < 1$:

    $x_{i^*,j} \leftarrow 1$;

    $z_j \leftarrow \max\{0,\ b_{i^*j}(1 - y_{i^*})\}$;

    $y_{i^*} \leftarrow y_{i^*} \cdot \left(1 + \frac{b_{i^*j}}{B_{i^*}}\right) + \frac{b_{i^*j}}{(c-1)B_{i^*}}$;

1. の成立

- 各 y_i は単調非減少
- 各ラウンドでは
  - $y_{i^*} \geq 1 \longrightarrow z_j = 0$ で実行可能
  - $y_{i^*} < 1 \longrightarrow z_j$ の定義より OK

双対問題

$$
\begin{array}{lll}
\min & \sum_{i \in N} B_i y_i + \sum_{j \in M} z_j & \\
\text{s.t.} & z_j \geq b_{ij}(1 - y_i) & (\forall i \in N, \forall j \in M) \\
 & \sum_{i \in N} x_{ij} \leq 1 & (\forall j \in M) \\
 & y_i \geq 0,\ z_j \geq 0 & (\forall i \in N,\ \forall j \in M)
\end{array}
$$

アルゴリズムの解析

次の 3 つが成り立つ

1. 双対問題について常に実行可能
2. 主問題についてほぼ実行可能
   各 i について予算超過は高々 1 回
3. $\Delta D \leq (1 + \frac{1}{c-1})\Delta P$

初期化: $y_i \leftarrow 0\ (\forall i \in [n])$;

Foreach j:

$i^* \in \arg\max_{i \in N}\{b_{ij}(1 - y_i)\}$ を選択;

if $y_{i^*} < 1$:

$x_{i^*,j} \leftarrow 1$;

$z_j \leftarrow \max\{0,\ b_{i^*j}(1 - y_{i^*})\}$;

$y_{i^*} \leftarrow y_{i^*} \cdot \left(1 + \frac{b_{i^*j}}{B_{i^*}}\right) + \frac{b_{i^*j}}{(c-1)B_{i^*}}$;

2. の成立

- $c = (1 + R_{\max})^{1/R_{\max}}$ なので
  $\sum_{j \in M} b_{ij}x_{ij} \geq B_i \Rightarrow y_i \geq 1$ が成立

  $c^x \leq 1 + x\ (\forall x \in [0, R_{\max}])$ を用いると

  $y_i \geq \frac{1}{c-1}\left(c^{\sum_{j \in M} b_{ij}x_{ij}/B_i} - 1\right)$ が帰納法により成立するから

- よって各 i に対して,
  予算制約を破るのは高々 1 回

主問題

$$\begin{array}{lll} \max & \sum_{j \in M}\sum_{i \in N} b_{ij}x_{ij} & \\ \text{s.t.} & \sum_{j \in M} b_{ij}x_{ij} \leq B_i & (\forall i \in N) \\ & x_{ij} \geq 0 & (\forall i \in N, \forall j \in M) \end{array}$$

アルゴリズムの解析

次の 3 つが成り立つ

1. 双対問題について常に実行可能
2. 主問題についてほぼ実行可能
   各 i について予算超過は高々 1 回
3. $\Delta D \leq (1 + \frac{1}{c-1})\Delta P$

初期化: $y_i \leftarrow 0$ $(\forall i \in [n])$;

Foreach j:

$i^* \in \arg\max_{i \in N}\{b_{ij}(1 - y_i)\}$ を選択;

if $y_{i^*} < 1$:

$x_{i^*,j} \leftarrow 1$;

$z_j \leftarrow \max\{0,\ b_{i^*j}(1 - y_{i^*})\}$;

$y_{i^*} \leftarrow y_{i^*} \cdot \left(1 + \frac{b_{i^*j}}{B_{i^*}}\right) + \frac{b_{i^*j}}{(c-1)B_{i^*}}$;

3. の成立

- 変数が更新されるラウンド $(y_{i^*} < 1)$ だけ考えれば良い
- 変化量は次のようになるので成立
  - $\Delta D = B_{i^*}\Delta y_{i^*} + z_j = (1 + \frac{1}{c-1})b_{i^*j}$
  - $\Delta P = b_{i^*j}$

アルゴリズムの解析

次の 3 つが成り立つ

1. 双対問題について常に実行可能
2. 主問題についてほぼ実行可能
   各 i について予算超過は高々 1 回
3. $\Delta D \leq (1 + \frac{1}{c-1})\Delta P$

初期化: $y_i \leftarrow 0\ (\forall i \in [n])$;

Foreach j:

$i^* \in \arg\max_{i \in N}\{b_{ij}(1 - y_i)\}$ を選択;

if $y_{i^*} < 1$:

$x_{i^*,j} \leftarrow 1$;

$z_j \leftarrow \max\{0,\ b_{i^*j}(1 - y_{i^*})\}$;

$y_{i^*} \leftarrow y_{i^*} \cdot \left(1 + \frac{b_{i^*j}}{B_{i^*}}\right) + \frac{b_{i^*j}}{(c-1)B_{i^*}}$;

まとめ

- 広告主 i について予算制約を破ってしまった場合も，
  最後の 1 回分を割り引いた分の利益は確保できる

$$\sum_{j \in M} b_{i,j} y_{i,j} - \max_{j \in M} b_{i,j} \geq (1 - R_{\max}) \sum_{j \in M} b_{i,j} y_{i,j}$$

- よって競合比は

$$\left(1 + \frac{1}{c-1}\right)(1 - R_{\max}) = \left(1 + \frac{1}{(1 + R_{\max})^{\frac{1}{R_{\max}}} - 1}\right)(1 - R_{\max}) \xrightarrow{R_{\max} \to +0} \frac{e}{e-1}$$

不可能性 [Kalyanasundaram and Pruhs 2000; Mehta et al. 2007]

定理 任意の乱択アルゴリズムの競合比は $e/(e-1)$ 以上

証明は演習．以下のランダムインスタンスに対し Yao の原理を用いる．

$M := \lceil 1/R_{\max} \rceil$，$n \to \infty$ の極限を考える

- $\pi : [n] \to [n]$ はランダムパーミュテーション
- $B_1 = \cdots = B_n = M$
- $k \cdot M + 1 \le j \le (k+1) \cdot M$ $(k = 0, 1, \ldots, n-1)$ について，

$$b_{ij} = \begin{cases} 1 & (i \in \{\pi(1), \ldots, \pi(n-k)\}) \\ 0 & (\text{その他}) \end{cases}$$

拡張

- 各ユーザー j に ℓ_j 個の広告枠がある場合 [Buchbinder et al. 2007]
- 各ユーザー j に ℓ_j 個の広告枠がある場合について，
  一般化第二価格オークション で値段が決まる場合 [Goel et al. 2010]

> k 番目の枠を得た広告主は $k+1$ 番目の入札額を支払う
> (ただし高い入札額をつけた広告主を割り当てる訳ではない)

- 各ユーザー j に T_j 秒のビデオ広告枠があり，各広告主 i は t_{ij} 秒の動画広告をもつ場合 [Sumita et al. 2017]

定理

どの場合も競合比 $\frac{e}{e-1}$ が達成可能

アウトライン

復習: オンライン被覆問題

主問題（被覆問題）

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} c_i x_i & \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{n} a_{i,j} x_i \geq 1 & (\forall j \in [m]) \\ & x_i \geq 0 & (\forall i \in [n]) \end{aligned}$$

双対問題（詰込問題）

$$\begin{aligned} \max \quad & \textstyle\sum_{j=1}^{m} y_j & \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{m} a_{i,j} y_j \leq c_i & (\forall i \in [n]) \\ & y_j \geq 0 & (\forall j \in [m]) \end{aligned}$$

- ラウンド j では j 番目の制約が与えられる
- どのラウンドでも与えられた制約を全て満たしていなければならない
- (主問題の) 変数は増加させることしかできない

定理 [Buchbinder and Naor 2005]

オンライン主双対法により競合比 $O(\log n)$ のアルゴリズムを設計可能

オンライン半正定値計画問題

主問題（被覆問題）

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{n} A_i x_i \succeq B_j \\ & x_i \geq 0 \qquad (\forall i \in [n]) \end{aligned}$$

双対問題（詰込問題）

$$\begin{aligned} \max \quad & B_j \bullet Y \\ \text{s.t.} \quad & A_i \bullet Y \leq c_i \qquad (\forall i \in [n]) \\ & Y \succeq O \end{aligned}$$

- 各ラウンドでは B が増加する（$B_j \succeq B_{j-1}$）
- どのラウンドでも与えられた制約を満たしていなければならない
- (主問題の) 変数は増加させることしかできない

定理 [Elad, Kale, and Naor 2016]

オンライン主双対法により競合比 $O(\log n)$ のアルゴリズムを設計可能

オンライン半正定値計画問題

主問題（被覆問題）

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{n} A_i x_i \succeq B_j \\ & x_i \geq 0 \qquad (\forall i \in [n]) \end{aligned}$$

双対問題（詰込問題）

$$\begin{aligned} \max \quad & B_j \bullet Y \\ \text{s.t.} \quad & A_i \bullet Y \leq c_i \qquad (\forall i \in [n]) \\ & Y \succeq O \end{aligned}$$

- 各ラウンドでは B が増加する（$B_j \succeq B_{j-1}$）
- どのラウンドでも与えられた制約を満たしていなければならない
- (主問題の) 変数は増加させることしかできない

定理 [Elad, Kale, and Naor 2016]

オンライン主双対法により競合比 $O(\log n)$ のアルゴリズムを設計可能

単調で凸な目的関数をもつオンライン被覆問題

主問題（被覆問題）

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Ax \geq 1 \\ & x \geq 0 \end{aligned}$$

双対問題（詰込問題）

$$\begin{aligned} \max \quad & \textstyle\sum_{j=1}^{m} y_j - f^*(A^\top y) \\ \text{s.t.} \quad & y \geq 0 \end{aligned}$$

$f^*(z) = \sup_{x \geq 0}(z^\top x - f(x))$

- ラウンド j では j 番目の制約が与えられる（$A \in \mathbb{R}_+^{m \times n}$）
- どのラウンドでも与えられた制約を全て満たしていなければならない
- (主問題の) 変数は増加させることしかできない

定理 [Azar et al. 2016]

オンライン主双対法により競合比 $O(p \log n)^p$ のアルゴリズムを設計可能
（ただし $p = \sup_{x \geq 0} \langle \nabla f(x),\, x \rangle / f(x)$）

単調で凸な目的関数をもつオンライン被覆問題

主問題（被覆問題）

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Ax \geq 1 \\ & x \geq 0 \end{aligned}$$

双対問題（詰込問題）

$$\begin{aligned} \max \quad & \textstyle\sum_{j=1}^{m} y_j - f^*(A^\top y) \\ \text{s.t.} \quad & y \geq 0 \end{aligned}$$

$f^*(z) = \sup_{x \geq 0}(z^\top x - f(x))$

- ラウンド j では j 番目の制約が与えられる（$A \in \mathbb{R}_+^{m \times n}$）
- どのラウンドでも与えられた制約を全て満たしていなければならない
- (主問題の) 変数は増加させることしかできない

定理 [Azar et al. 2016]

オンライン主双対法により競合比 $O(p \log n)^p$ のアルゴリズムを設計可能
（ただし $p = \sup_{x \geq 0} \langle \nabla f(x),\, x \rangle / f(x)$）

ML prediction

入力予想付きの性能指標 [Lykouris and Vassilvitskii 2018]

予想の信用度

入力予想 $\mathcal{A}$ の下で ALG が $C(\lambda)$-consistent, $R(\lambda)$-robust $(0 < \lambda \leq 1)$ とは

$$\mathrm{obj}(\mathrm{ALG}(\sigma, \mathcal{A}, \lambda)) \leq \min\left\{ C(\lambda) \cdot S(\sigma, \mathcal{A}),\ R(\lambda) \cdot \mathrm{obj}(\mathrm{OPT}(\sigma)) \right\}$$

予想を盲目的に信じるアルゴリズムのコスト

オンライン主双対法は予想付きの場合に拡張可能 [Bamas, Maggiori, Svensson 2020]

購入費用 B のスキーレンタル問題

$$\mathrm{obj}(\mathrm{ALG}(\sigma, N, \lambda)) \leq \min\left\{ \frac{\lambda}{1-(1+\frac{1}{B})^{-\lambda B}} \cdot S(\sigma, N),\ \frac{1}{1-(1+\frac{1}{B})^{-\lambda B}} \cdot \mathrm{obj}(\mathrm{OPT}(\sigma)) \right\}$$

N 回スキーに行くと予想

$N \leq B$ ならずっとレンタル，$N > B$ なら初回に購入でのコスト

集合被覆問題

$$\mathrm{obj}(\mathrm{ALG}(\sigma, \mathcal{A}, \lambda)) \leq \min\left\{ O\left(\frac{1}{1-\lambda}\right) \cdot S(\sigma, \mathcal{A}),\ O\left(\log \frac{n}{\lambda}\right) \cdot \mathrm{obj}(\mathrm{OPT}(\sigma)) \right\}$$

解の予想

$\mathcal{A}$ を選択した時のコスト

入力モデル

最悪ケースを考えるのは悲観的すぎる．平均ケースは？

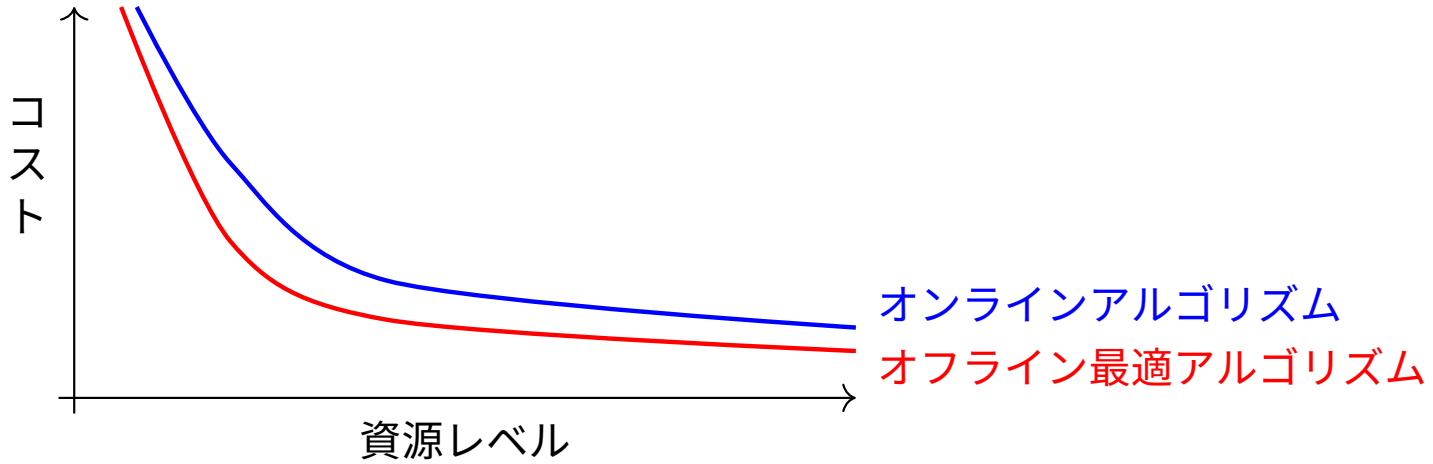
- 未知の独立同一分布
- 既知の独立同一分布
- 既知の同一とは限らない分布 (預言者の不等式)
- ランダム順序モデル (秘書問題)

- 最悪ケースと平均ケース両方での性能を保証

資源拡大モデル

比較相手として単純なオフライン最適解は妥当？

- アドバーサリにハンデをつけてアルゴリズムの評価指標を緩くする
- 資源拡大モデルの例
  - ページング問題におけるキープできるページの数
  - k サーバー問題におけるサーバー数
  - スケジューリング問題におけるマシンの処理速度や台数
  - ナップサック問題におけるナップサックの容量
  - ビンパッキング問題におけるビンの容量



まとめ

オンライン最適化の競合比解析について，
「双対性」を利用したアルゴリズムの設計手法，不可能性の証明手法を紹介

オンライン主双対法

- 元問題の解 と 双対問題の実行可能解 を同時に構成
- 各ラウンドでの目的関数値の変化の比を ρ で抑える
  ⟶ 最終的な値の比も ρ 以下 ⟶ アルゴリズムは競合比 ρ

Yao の原理

任意の入力の分布 $\hat{\boldsymbol{p}}$ について

$$\inf_{\boldsymbol{q}} \sup_{\sigma} \mathbb{E}_{y\sim\boldsymbol{q}}[\mathrm{cost}(\mathrm{ALG}_y(\sigma))] \geq \inf_{\mathrm{ALG}} \mathbb{E}_{x\sim\hat{\boldsymbol{p}}}[\mathrm{cost}(\mathrm{ALG}(\sigma_x))]$$

さらにオンライン最適化の競合比解析を学ぶには