１４．プライマルデュアル法

線形計画法の双対の概念を応用した近似アルゴリズムがある。プライマルデュアル法（主双対法）という。

１４．１　LP双対性入門

まず、次のような線形計画問題を考える。

問題A（主問題）

目的関数

$$\min \quad f(x) = 7x_1 + x_2 + 5x_3$$

制約条件

$$x_1 - x_2 + 3x_3 \geq 10 \quad \cdots(1)$$

$$5x_1 + 2x_2 - x_3 \geq 6 \quad \cdots(2)$$

$$x_1, x_2, x_3 \geq 0$$

問題AのようなかたちをLPの標準形という。

制約条件を満たす解を、実行可能解という。

例えば、

$$\boldsymbol{x} = (2, 1, 3)$$

は実行可能解。

実際、

$$1 \bullet 2 - 1 \bullet 1 + 3 \bullet 3 = 10 \geq 10 \quad \cdots (1)$$

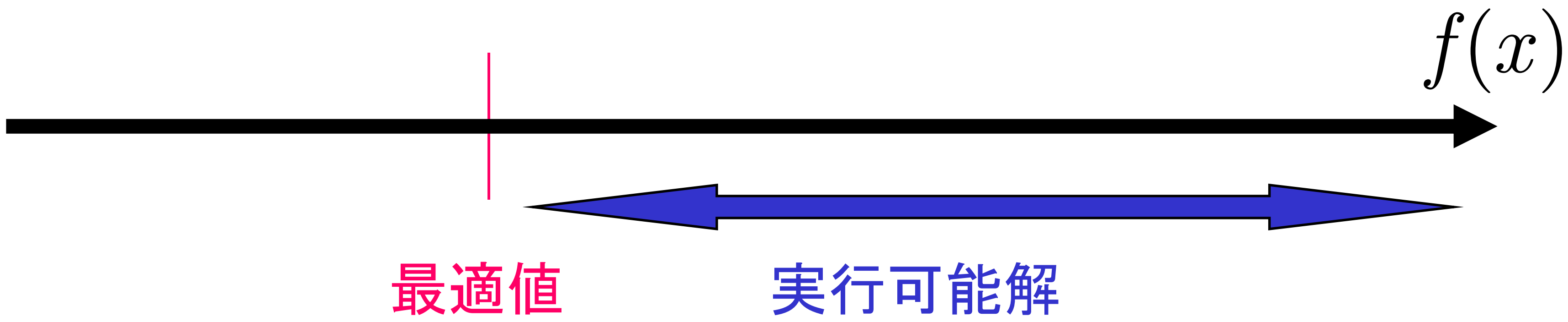
$$5 \bullet 2 + 2 \bullet 1 - 1 \bullet 3 = 9 \geq 6 \quad \cdots (1)$$

このとき、目的関数値は、

$$f(x) = 7 \bullet 2 + 1 \bullet 1 + 5 \bullet 3 = 30$$

最適値は実行関数値以下であるので、この場合30以下になる。

このことは実行可能関数値は、最適値の上界を与えているとみなせる。



ここで、下界について考える。

係数から、(1)式と(2)式の和をとると、目的関数値の下界を導くことができる。

$$f(x) = 7x_1 + x_2 + 5x_3$$

$$\geq 6x_1 + x_2 + 2x_3$$

$$= (x_1 - x_2 + 3x_3) + (5x_1 + 2x_2 - x_3)$$

$$\geq 10 + 6 = 16$$

x_3

$$\therefore f(x) \geq 16$$

このように、制約式に非負数を乗じて加えると、目的関数の下界が得られる。この非負数に注目してもう一つのLP問題を得ることができる。下解はできるだけ大きくするほうが良いことに注意うする。

問題B（双対問題）

目的関数

$$\max \quad f^*(\boldsymbol{y}) = 10y_1 + 6y_2$$

制約条件

$$y_1 + 5y_2 \leq 7 \qquad \cdots(1)^*$$

$$-y_1 + 2y_2 \leq 1 \qquad \cdots(2)^*$$

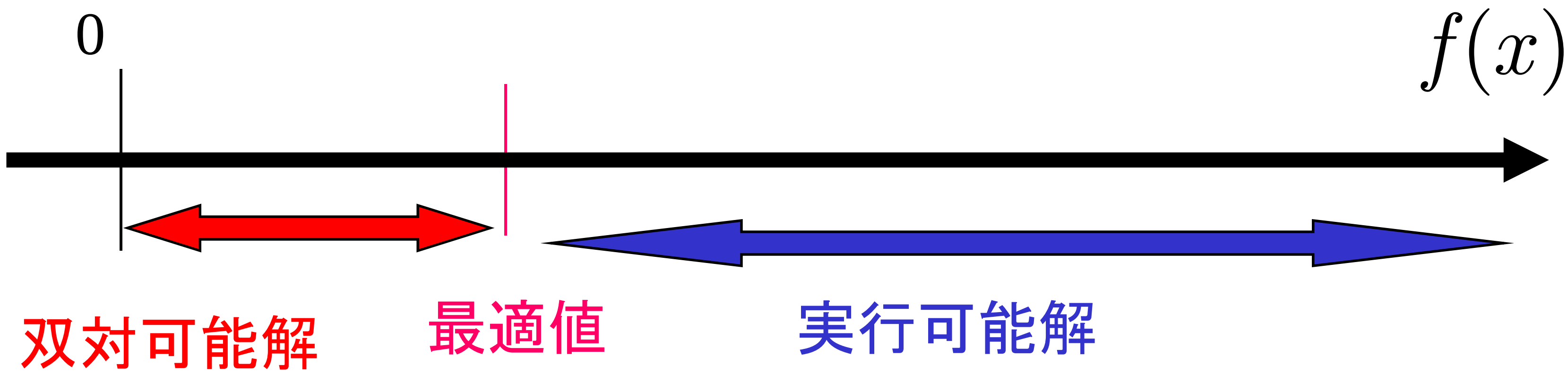
$$3y_1 - y_2 \leq 5 \qquad \cdots(3)^*$$

$$y_1, y_2 \geq 0$$

制約式の不等号の向きが変化しないように、
非負の条件が必要である。

このようにして得られたLP問題を、元の問題に対する
双対問題（dual,デュアル）という。

双対問題に対して、元の問題を主問題（primal,プライマル）
という。



双対関係

問題A（主問題）

目的関数

$$\min \quad f(x) = \sum_{j=1}^{n} c_j x_j$$

制約条件

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i \qquad (i = 1, \cdots, m)$$

$$x_j \geq 0 \qquad (j = 1, \cdots, n)$$

問題B（双対問題）

目的関数

$$\max \quad f^*(\boldsymbol{y}) = \sum_{i=1}^{m} b_i y_i$$

制約条件

$$\sum_{i=1}^{m} a_{ij} y_i \leq c_j \qquad (j = 1, \cdots, n)$$

$$y_j \geq 0 \qquad (i = 1, \cdots, m)$$

双対問題の双対問題は、主問題になる。

弱双対定理

弱双対定理

$\boldsymbol{x} = (x_1, \cdots, x_n)$ と $\boldsymbol{y} = (y_1, \cdots, y_m)$ がそれぞれ、主問題と双対問題の実行可能解のとき、次式が成り立つ。

$$f(\boldsymbol{x}) \geq f^*(\boldsymbol{y})$$

$$\therefore \sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i$$

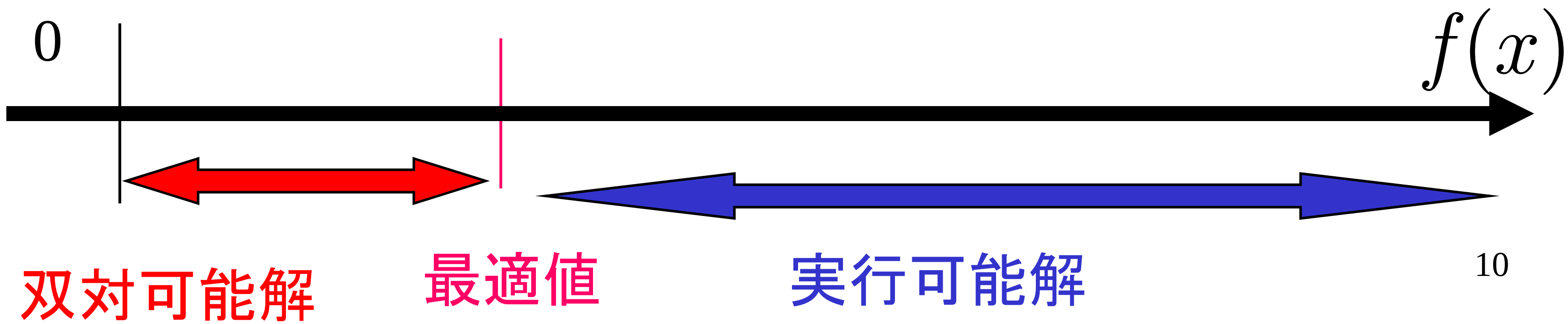
双対定理

双対定理

$\boldsymbol{x}^* = (x^*_1, \cdots, x^*_n)$ と $\boldsymbol{y}^* = (y^*_1, \cdots, y^*_m)$ をそれぞれ、主問題と双対問題の最適解のとき、次式が成り立つ。

$$f(\boldsymbol{x}^*) = f^*(\boldsymbol{y}^*)$$

$$\therefore \sum_{j=1}^{n} c_j x^*_j = \sum_{i=1}^{m} b_i y^*_i$$



相補条件

相補条件

$\boldsymbol{x} = (x_1, \cdots, x_n)$ と $\boldsymbol{y} = (y_1, \cdots, y_m)$ がそれぞれ、主問題と双対問題の実行可能解のとき、次式が成り立つ。

このとき、$\boldsymbol{x}$ と $\boldsymbol{y}$ が最適解であるための必要十分条件は、以下の条件が成立することである。

(1)主相補条件:各 $1 \leq j \leq n$ に対して、

$$x_j = 0 \quad \text{あるいは} \quad \sum_{i=1}^{m} a_{ij} y_i = c_j$$ である。

(2)双対相補条件:各 $1 \leq i \leq m$ に対して、

$$y_j = 0 \quad \text{あるいは} \quad \sum_{j=1}^{n} a_{ij} x_j = b_i$$ である。

14．2 プライマルデュアル法

相補条件を緩和することにより、近似アルゴリズムが得られる。

緩和相補条件

（1）主相補条件: $\alpha \geq 1$ とする。
各 $1 \leq j \leq n$ に対して、$x_j = 0$ あるいは

$$\frac{1}{\alpha} c_j \leq \sum_{i=1}^{m} a_{ij} y_i \leq c_j$$

である。

（2）双対相補条件: $\beta \geq 1$ とする。
各 $1 \leq i \leq m$ に対して、$y_j = 0$ あるいは

$$b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq \beta \bullet b_i$$

である。

プライマルデュアル法の近似率

近似率

緩和した相補条件を共に満足するとき、
以下が成り立つ。

$$\sum_{j=1}^{n} c_j x_j \leq (\alpha \bullet \beta) \sum_{i=1}^{m} b_i y_i$$

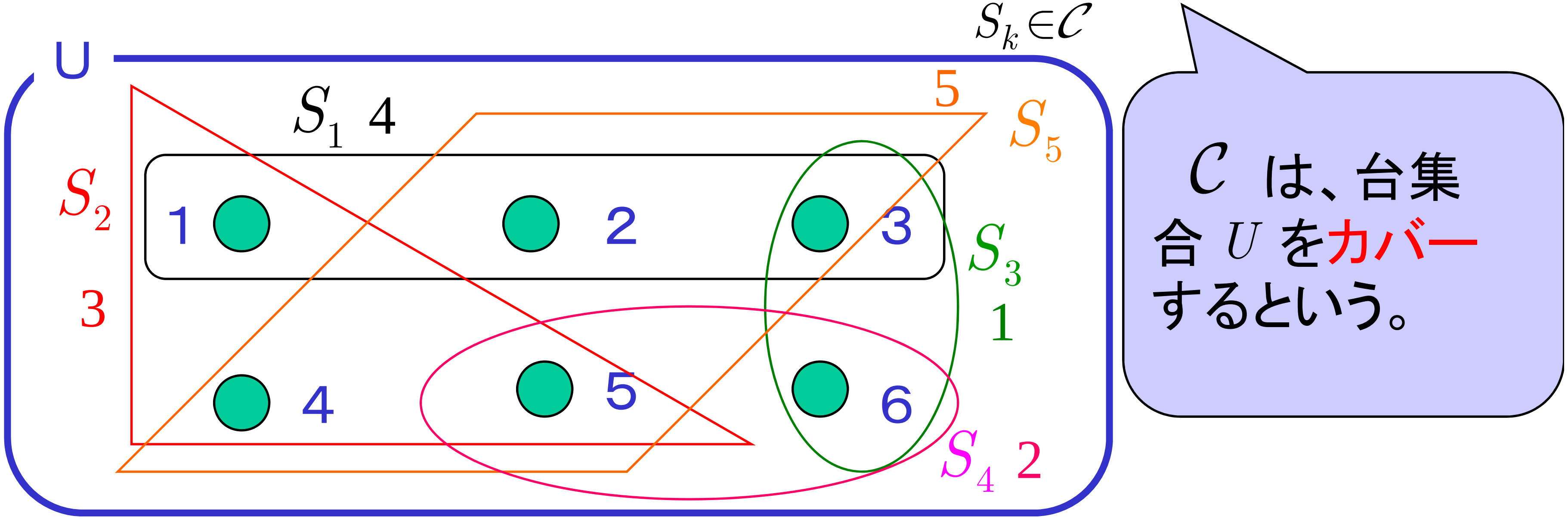
$$\therefore f^*(\boldsymbol{y}) \leq f(\boldsymbol{x}) \leq (\alpha \bullet \beta) f^*(\boldsymbol{y})$$

よって、 $\alpha \bullet \beta$ 近似アルゴリズムである。

14．3　集合カバー

ある集合 $U = \{e_1, \cdots, e_n\}$（台集合という）と、その部分集合からなる族　$\mathcal{S} = \{S_1, S_2, \cdots, S_m\}, S_i \subseteq U$ が与えられたとき、族の中のいくつかの集合を選んで、その和集合が台集合をふくむようにする。さらに、この族の各要素には、コストが割り当てられている。このとき、次の式を満たすような部分族 $\mathcal{C}$ でコスト最小のものを求める。

$$\mathcal{C} = \{S_p, \cdots, S_q\} \subseteq \mathcal{S}, U \subseteq \bigcup_{S_k \in \mathcal{C}} S_k$$



集合カバーの難しさ

集合カバー問題は、NP完全である。

証明略

集合カバーの整数計画法による定式化

コスト関数を $c : \mathcal{S} \rightarrow \mathbb{Q}^{+}$ とする。

目的関数

$$\min \quad f(x) = \sum_{S \in \mathcal{S}} c(S) x_S$$

制約条件

$$\sum_{S : e \in S} x_S \geq 1 \quad (e \in U)$$

$$x_S \in \{0, 1\} \quad (S \in \mathcal{S})$$

↓ 線形緩和

集合カバーの緩和線形計画法

目的関数

$$\min \quad f(x) = \sum_{S \in \mathcal{S}} c(S) x_S$$

制約条件

$$\sum_{S : e \in S} x_S \geq 1 \quad (e \in U)$$

$$x_S \geq 0 \quad (S \in \mathcal{S})$$

標準形になっている。
少数集合カバー

双対問題

少数集合カバーの双対問題

目的関数

$$\max \quad f^*(\boldsymbol{y}) = \sum_{e \in U} y_e$$

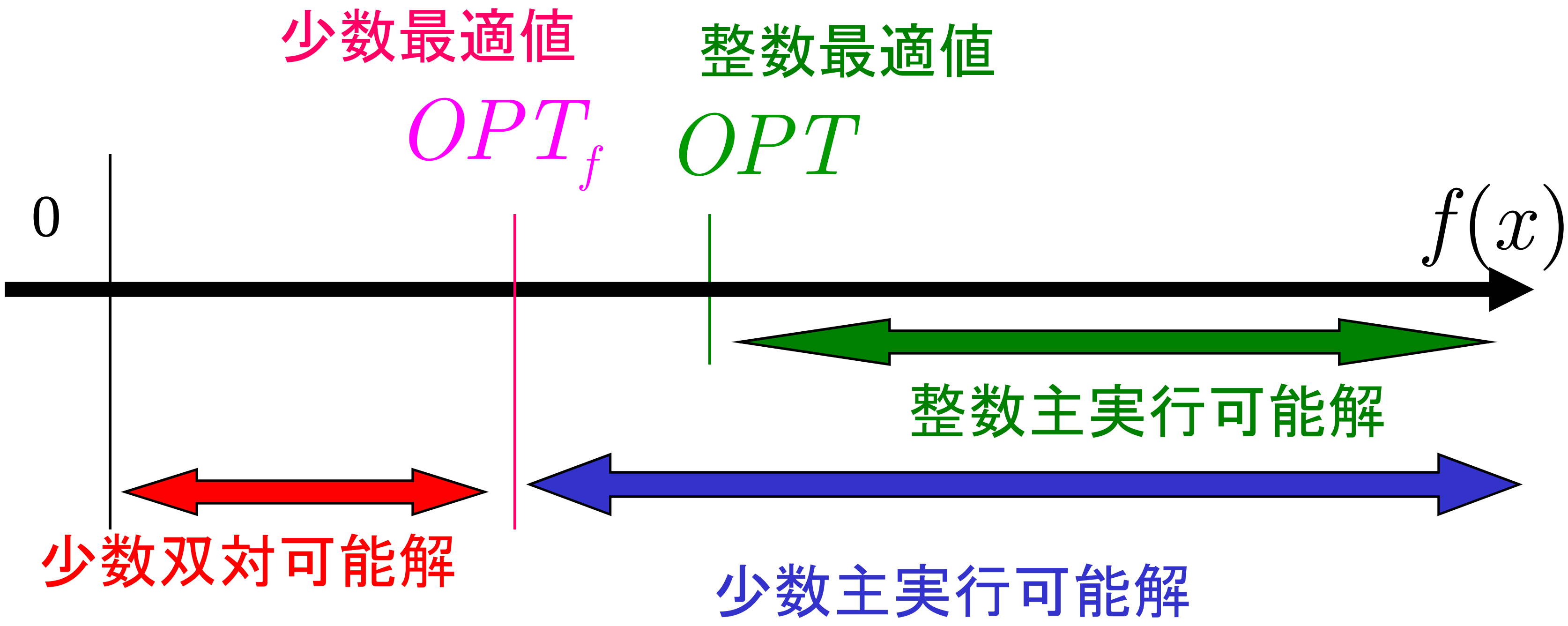
制約条件

$$\sum_{e \in U} y_e \le c(S) \quad (S \in \boldsymbol{\mathcal{S}})$$

$$y_e \ge 0 \quad (e \in U)$$

各要素eに対応する”物質”を、集合Sに詰め込むことをイメージするとよい。直感的に、各集合Sには、ある一定以上の物質を詰め込むことができない。(この条件を、破ることを、オーバーパックということがある。)

関数値の関係



14．4　集合カバーへのプライマルデュアル法の適用

（1）主相補条件：全ての$S \in \mathcal{S}$ に対して、

$$x_S \neq 0 \quad \Rightarrow \quad \sum_{e:e\in S} y_e = c(S)$$

主相補条件は、厳密解の条件。

$$\therefore \alpha = 1$$

（2）双対相補条件：

$S \in \mathcal{S}$に対して、要素 $e \in U$ 頻度の最大値をkとする。このとき、

全ての $e \in U$ に対して、

$$y_e \neq 0 \quad \Rightarrow \quad \sum_{S:e\in S} x_S \leq 1 \bullet k$$

双対相補条件が、緩和されている。

$$\therefore \beta = k$$

相補条件の利用

集合Sは、

$$\sum_{e:e \in S} y_e = c(S)$$

を満たすとき、タイトと呼ばれる。

主問題の変数は整数性を保ちながら更新する。
しかも、タイトな集合のみから集合を集合カバーに選ぶ。

双対変数の値が非零要素のみ、 k 個までの集合
でカバーされる。

アルゴリズム

集合カバー(近似率 k)

1.（初期化） $\boldsymbol{x} \leftarrow \boldsymbol{0}; \boldsymbol{y} \leftarrow \boldsymbol{0};$

2. すべての要素がカバーされるまで以下を繰り返す。

2.1:カバーされていない要素を1つ選び e とし、e を含むどれかの集合がタイトになるまで y_e の値を増加する;

2.2:タイトな集合をすべてカバーに選んで、$\boldsymbol{x}$ を更新する;

2.3:これらの集合に含まれている要素は、カバーされているとする。

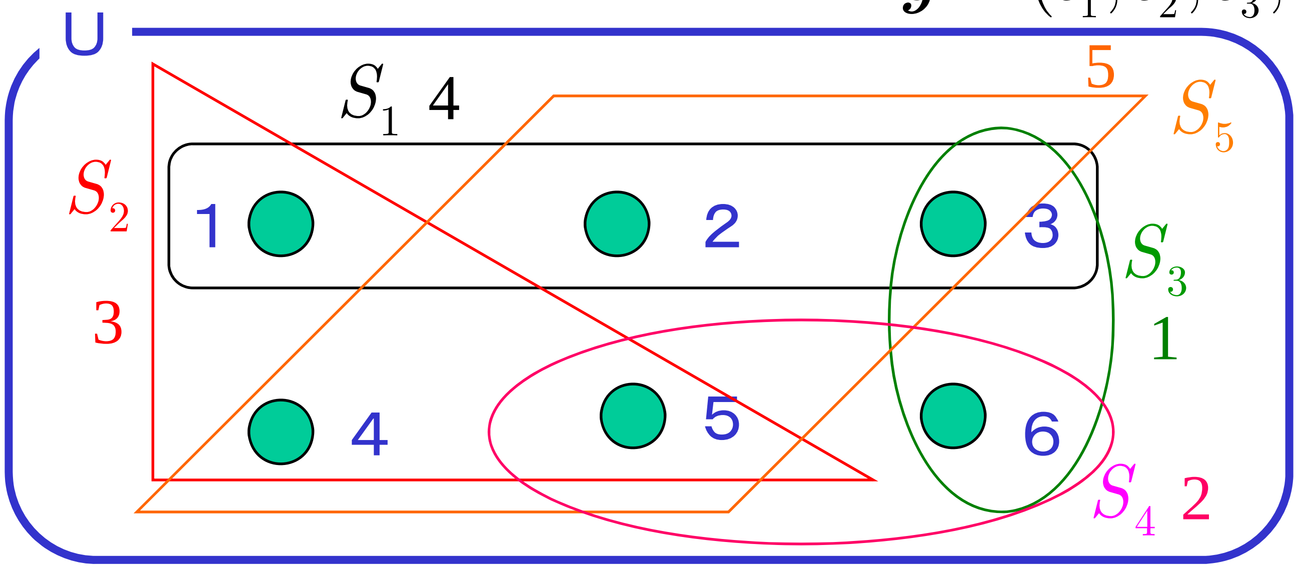
3. 集合カバーとして $\boldsymbol{x}$ を出力する。

アルゴリズムの動作例１

$\boldsymbol{x} = (S_1, S_2, S_3, S_4, S_5)$

$\boldsymbol{y} = (e_1, e_2, e_3, e_4, e_5, e_5)$

$k = 3$

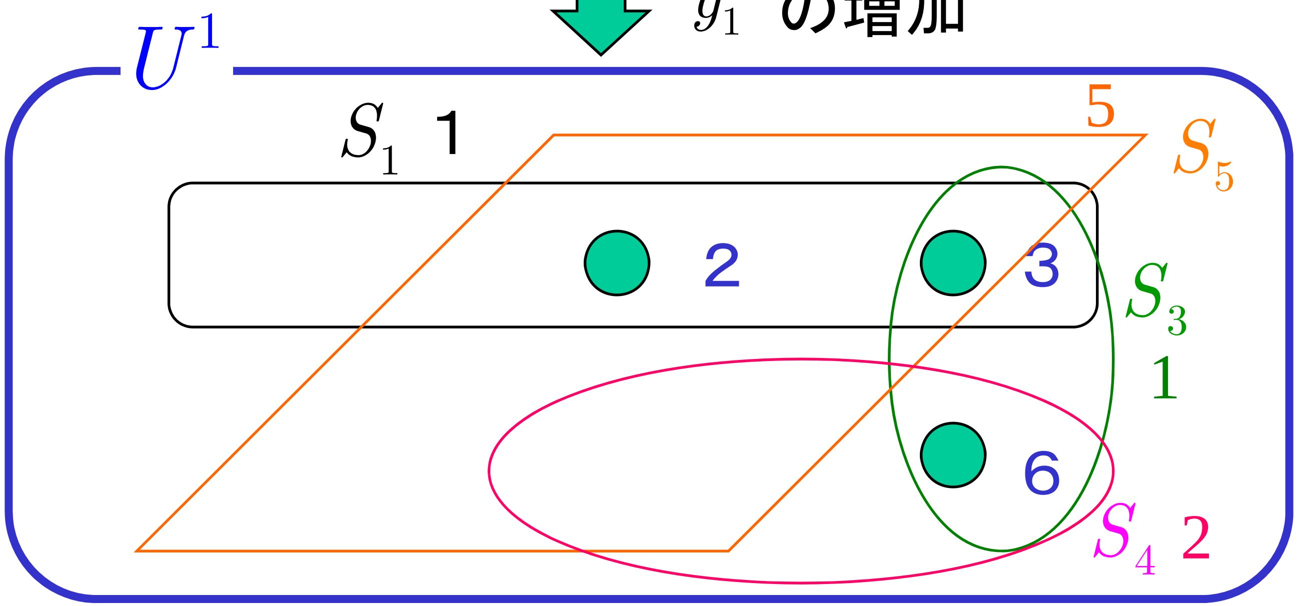


$\boldsymbol{x} = (0, 0, 0, 0, 0)$

$\boldsymbol{y} = (0, 0, 0, 0, 0, 0)$

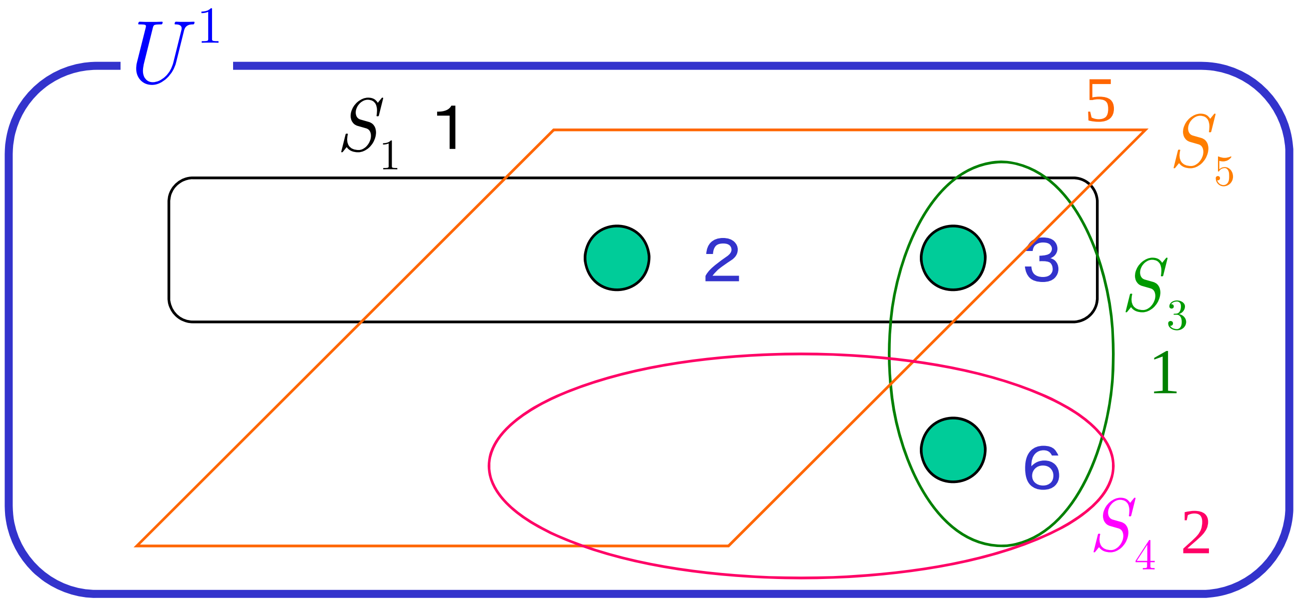
e_1 選択

y_1 の増加



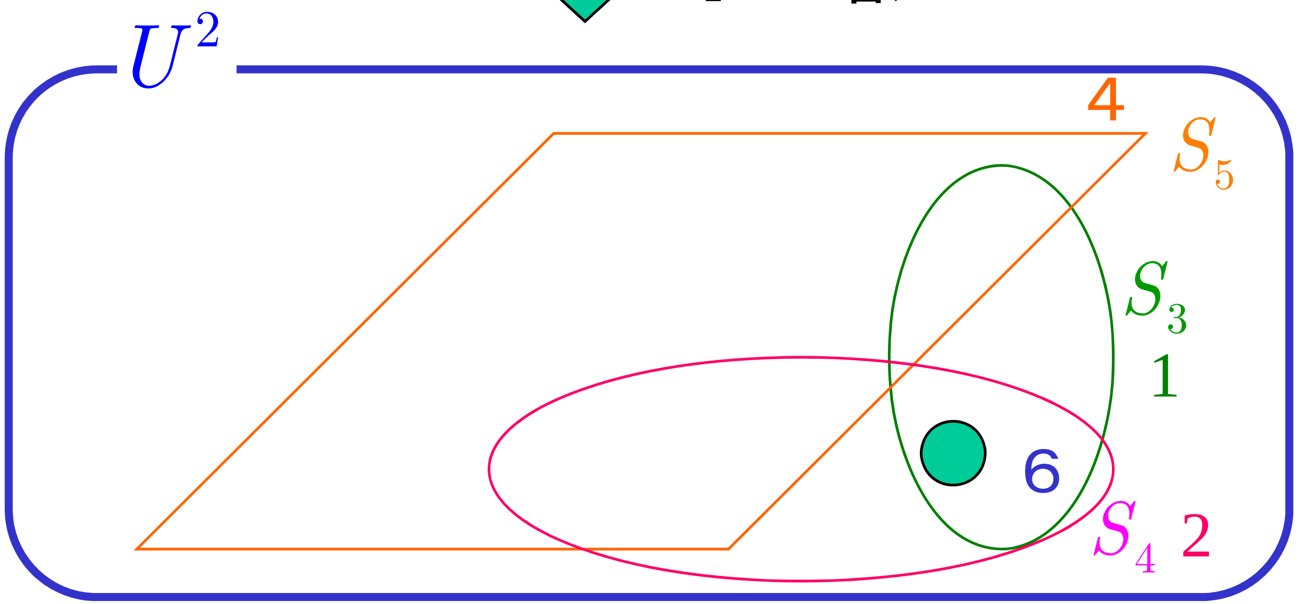
$\boldsymbol{x} = (0, 1, 0, 0, 0)$

$\boldsymbol{y} = (3, 0, 0, 0, 0, 0)$

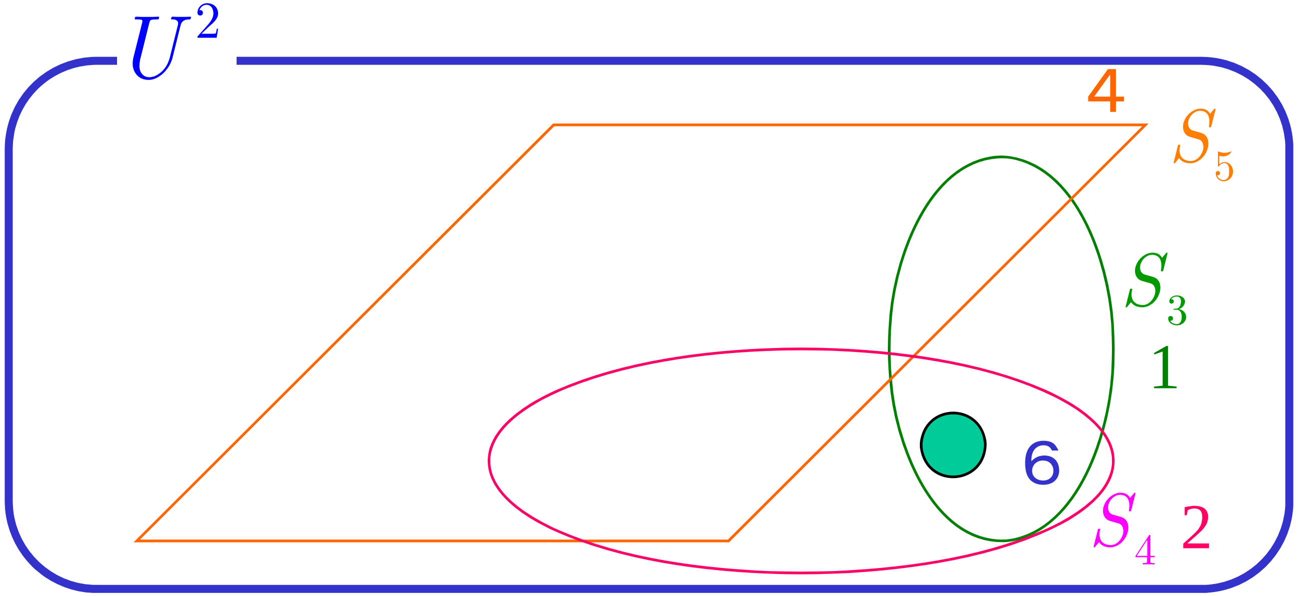


$$\boldsymbol{x} = (0, 1, 0, 0, 0)$$
$$\boldsymbol{y} = (3, 0, 0, 0, 0, 0)$$

e_2 選択
y_2 の増加



$$\boldsymbol{x} = (1, 1, 0, 0, 0)$$
$$\boldsymbol{y} = (3, 1, 0, 0, 0, 0)$$

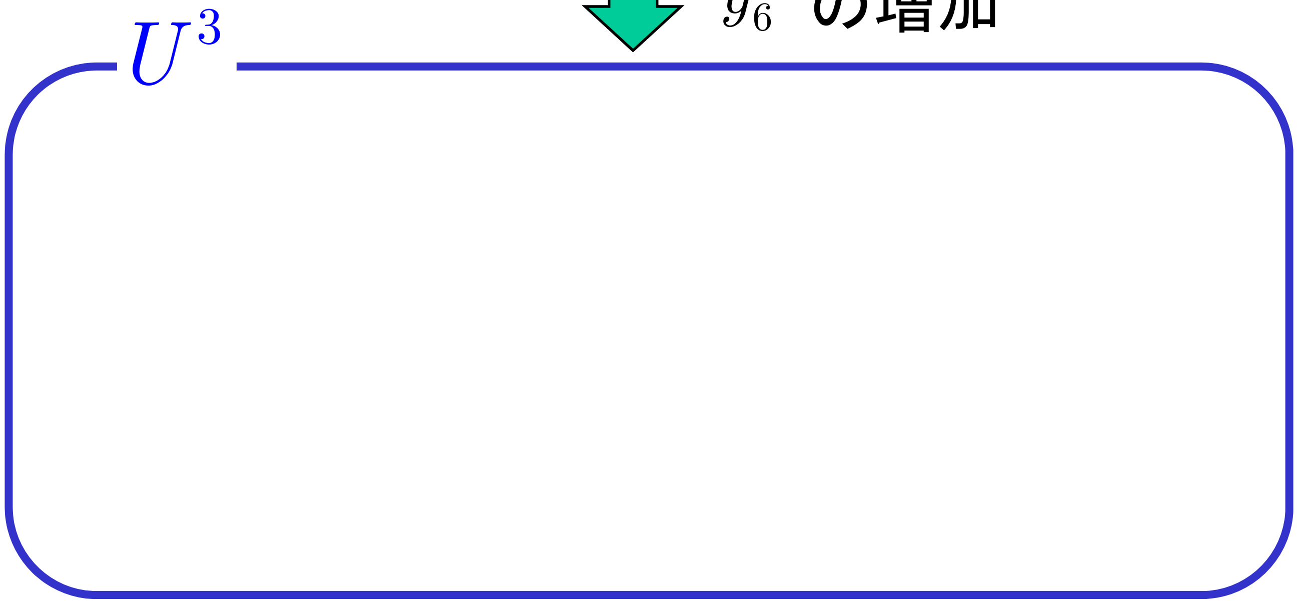


$$\boldsymbol{x} = (1, 1, 0, 0, 0)$$

$$\boldsymbol{y} = (3, 1, 0, 0, 0, 0)$$

e_6 選択

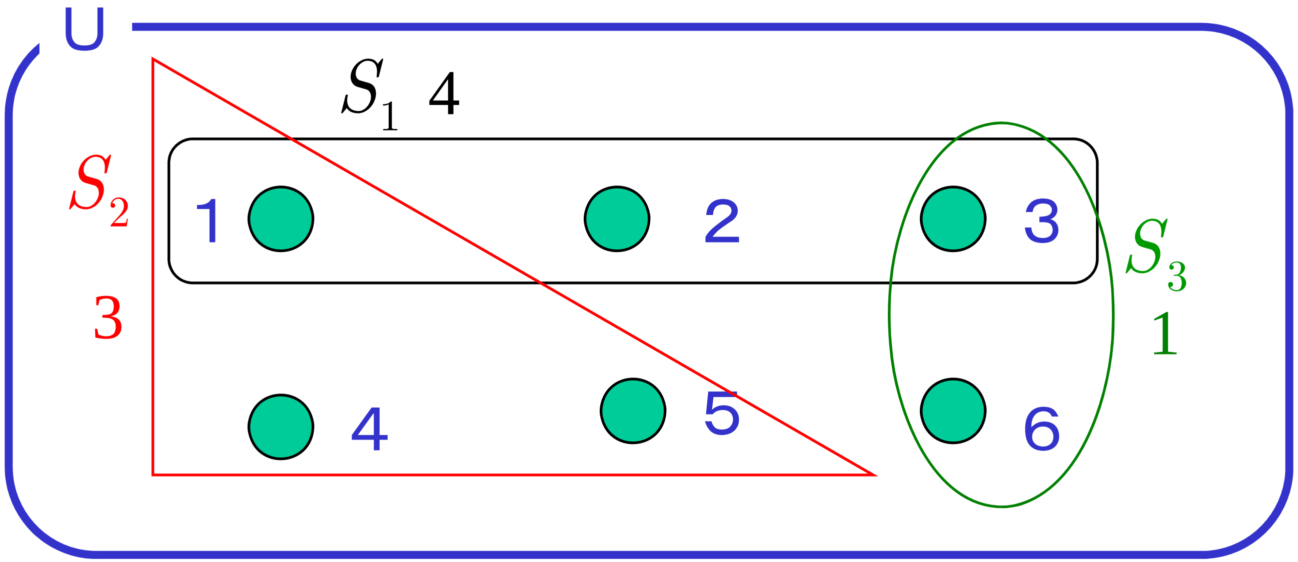
y_6 の増加



$$\boldsymbol{x} = (1, 1, 1, 0, 0)$$

$$\boldsymbol{y} = (3, 1, 0, 0, 0, 1)$$

解



$$\boldsymbol{x} = (1, 1, 1, 0, 0)$$

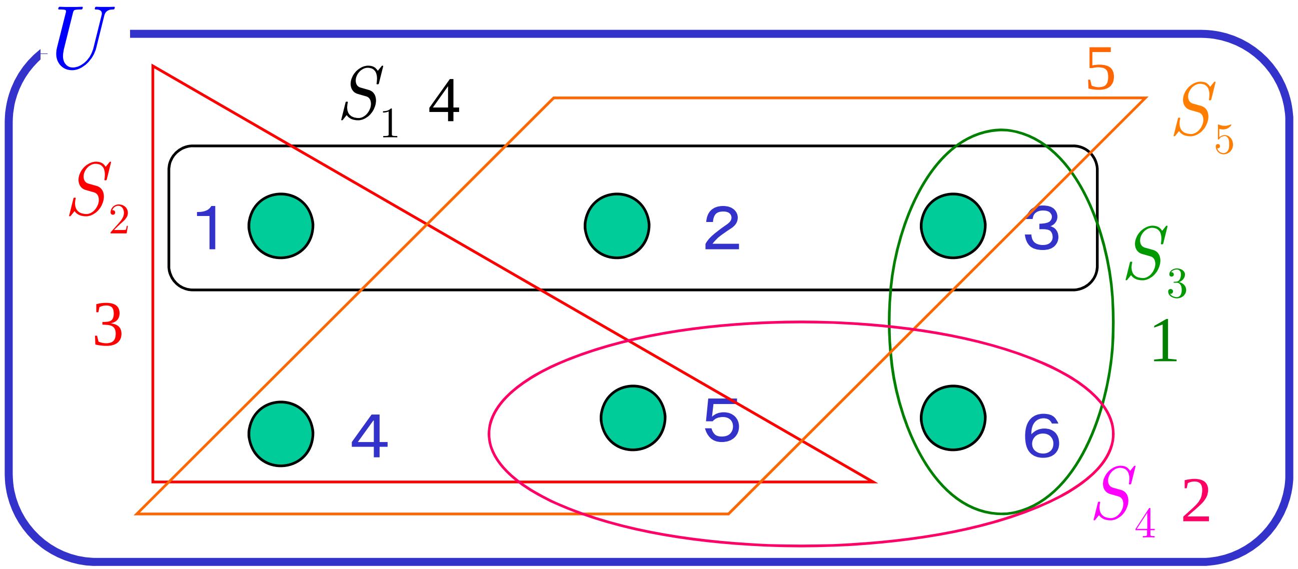
$$\boldsymbol{y} = (3, 1, 0, 0, 0, 1)$$

$$f(\boldsymbol{x}) = 4 + 3 + 1 = 8$$

$$f^*(\boldsymbol{y}) = 3 \bullet (3 + 1 + 1) = 15$$

アルゴリズムの動作例2

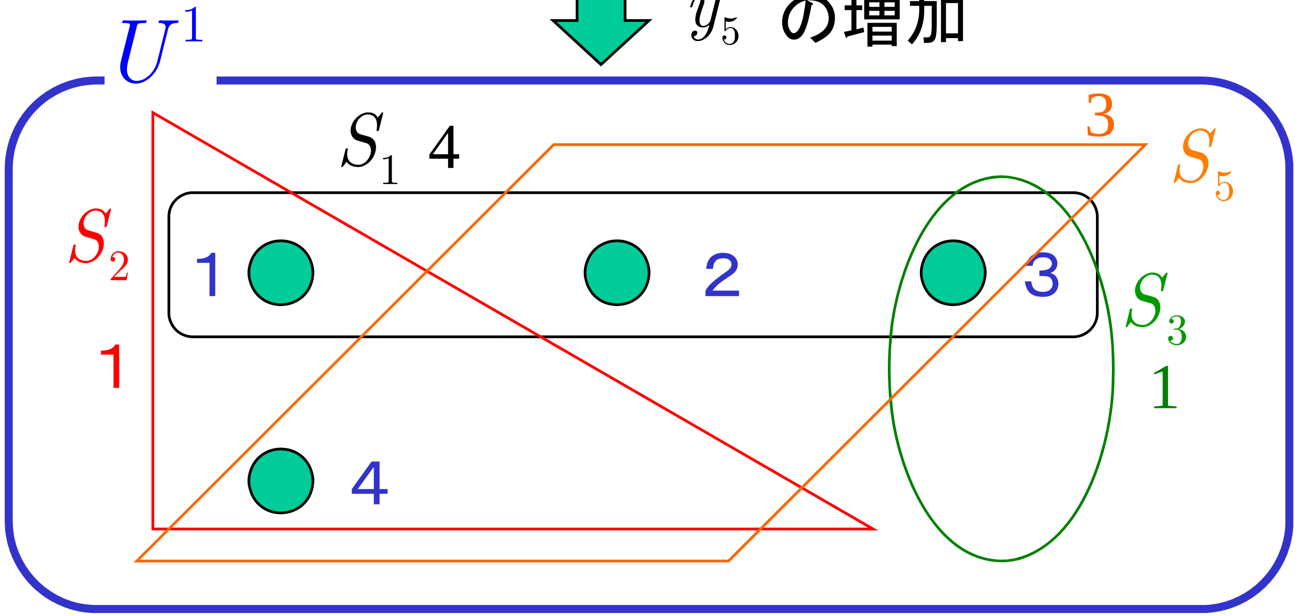
$k = 3$



$$\boldsymbol{x} = (0, 0, 0, 0, 0)$$

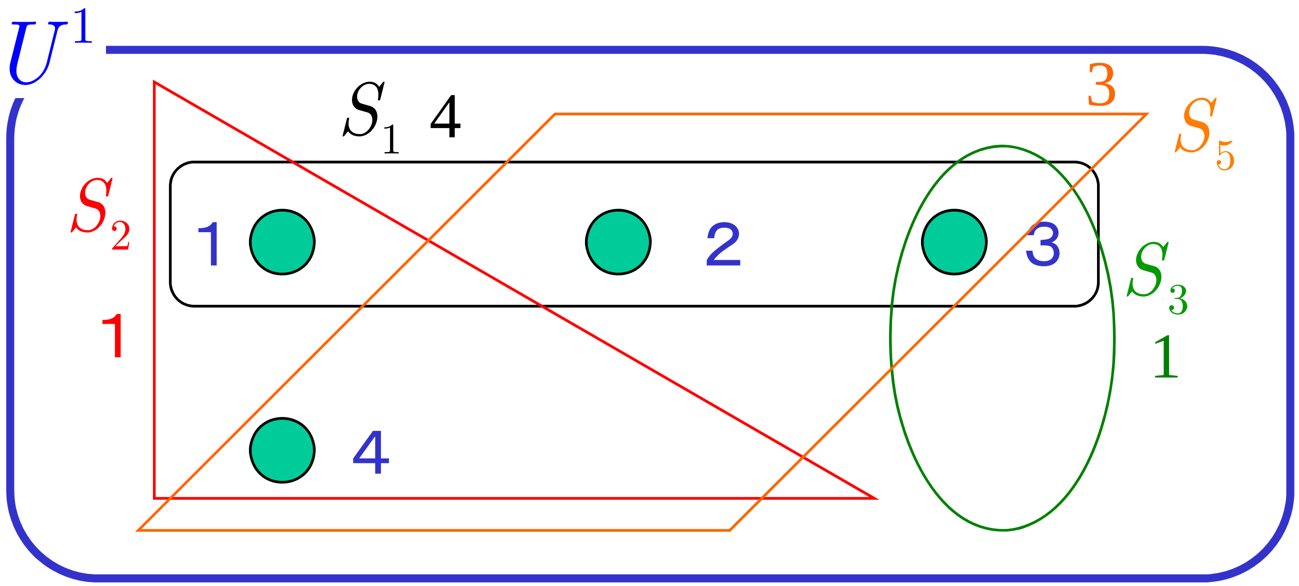
$$\boldsymbol{y} = (0, 0, 0, 0, 0, 0)$$

e_5 選択
y_5 の増加



$$\boldsymbol{x} = (0, 0, 0, 1, 0)$$

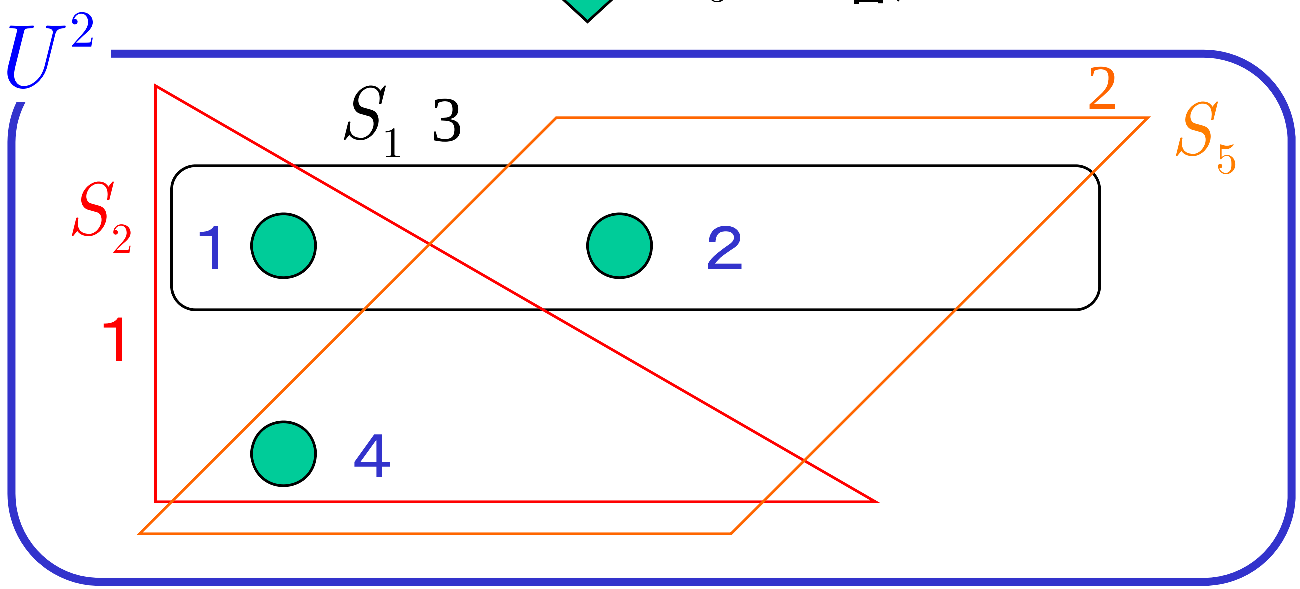
$$\boldsymbol{y} = (0, 0, 0, 0, 2, 0)$$



$$\boldsymbol{x} = (0, 0, 0, 1, 0)$$

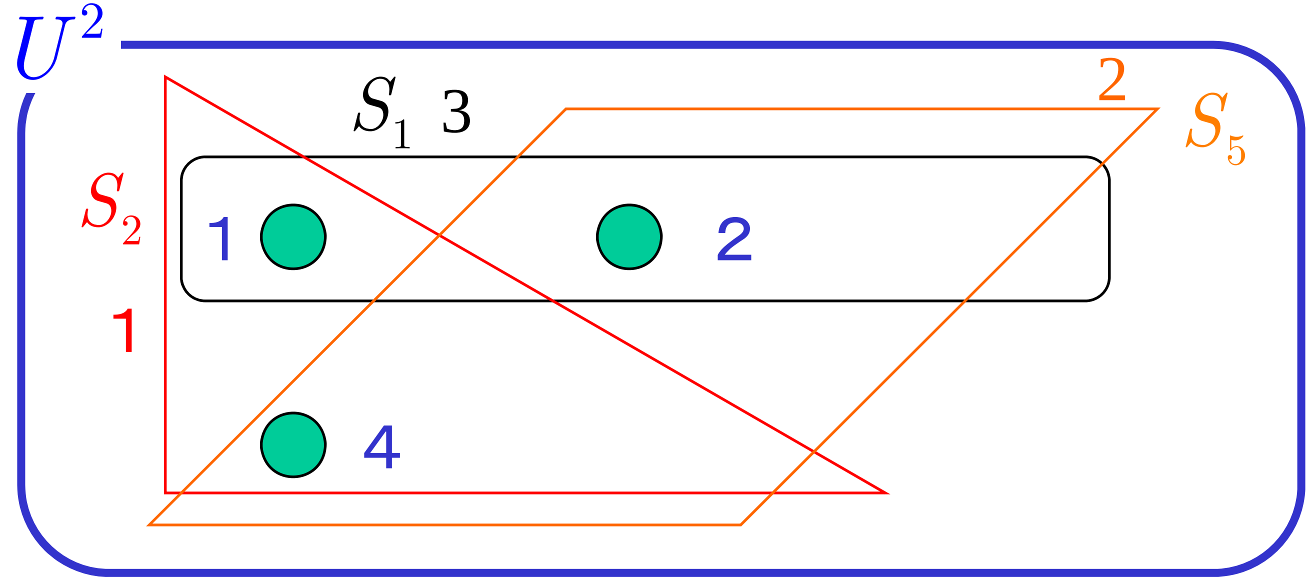
$$\boldsymbol{y} = (0, 0, 0, 0, 2, 0)$$

e_3 選択
y_3 の増加



$$\boldsymbol{x} = (0, 0, 1, 1, 0)$$

$$\boldsymbol{y} = (0, 0, 1, 0, 2, 0)$$

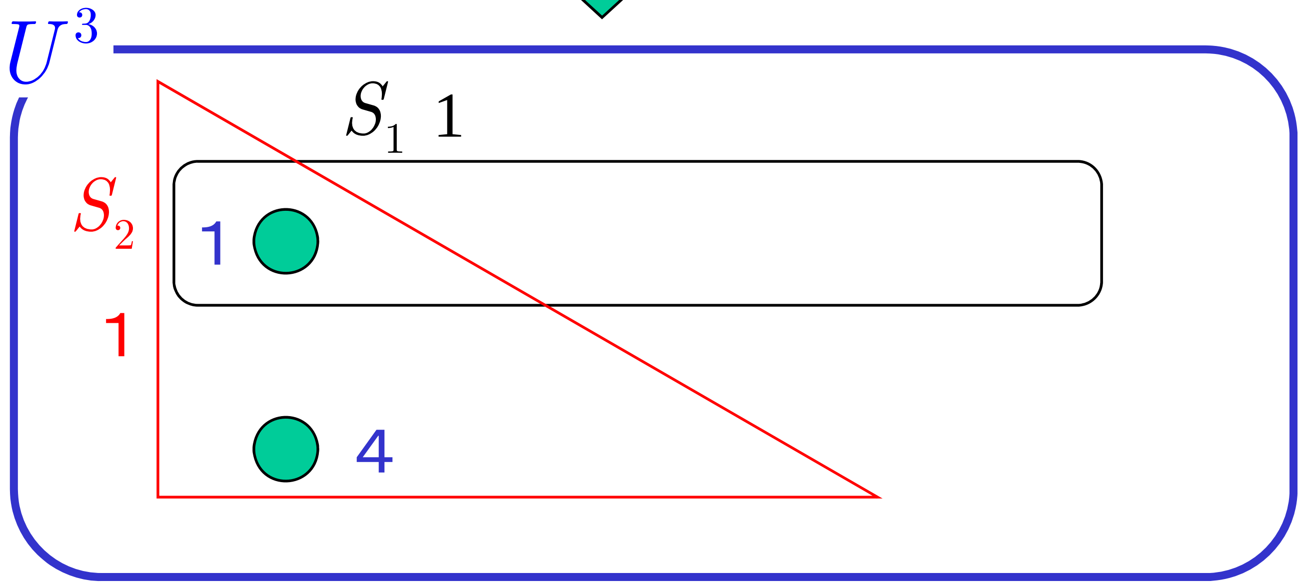


$$\boldsymbol{x} = (0, 0, 1, 1, 0)$$

$$\boldsymbol{y} = (0, 0, 1, 0, 2, 0)$$

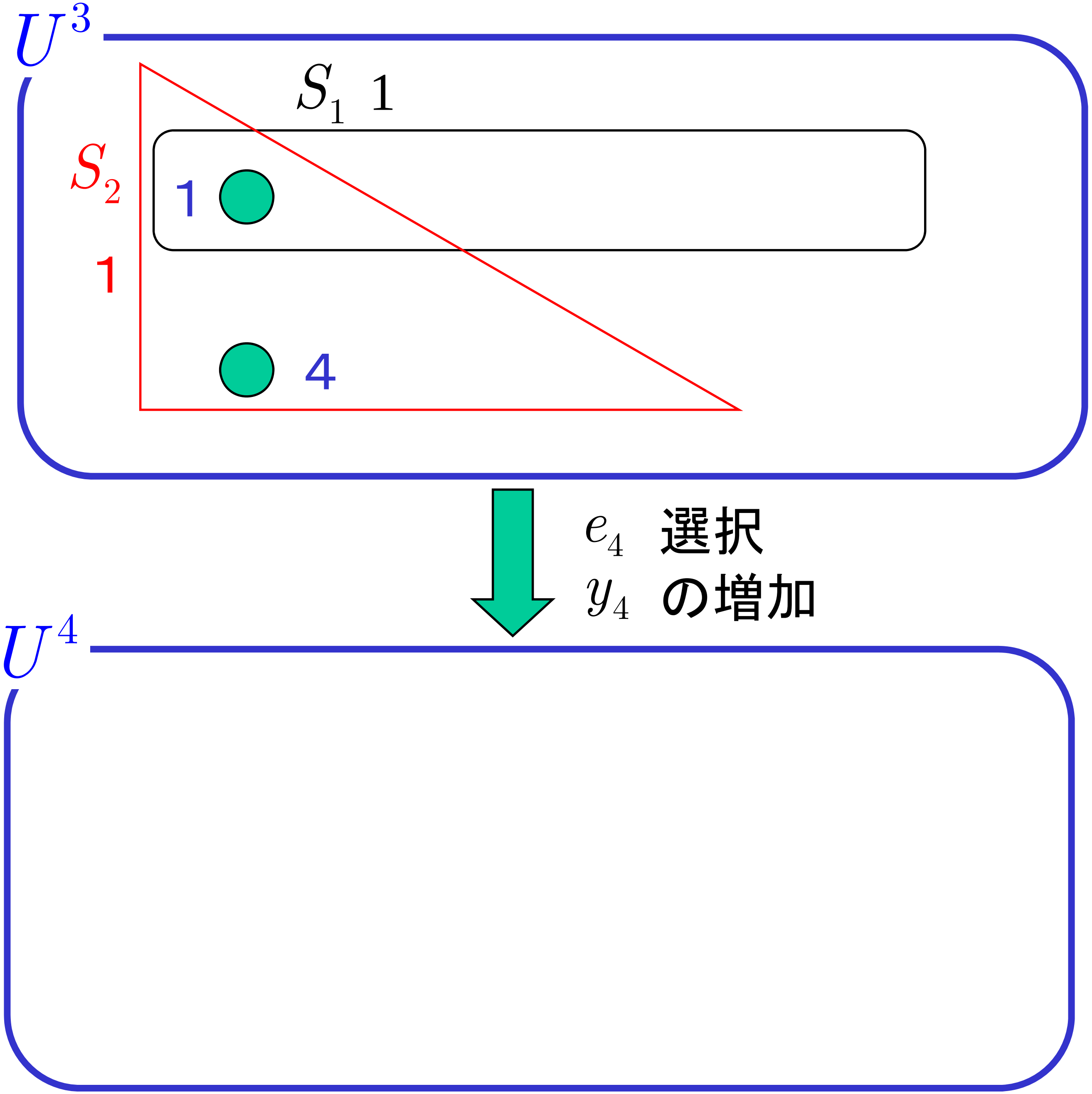
e_2 選択

y_2 の増加



$$\boldsymbol{x} = (0, 0, 1, 1, 1)$$

$$\boldsymbol{y} = (0, 2, 1, 0, 2, 0)$$



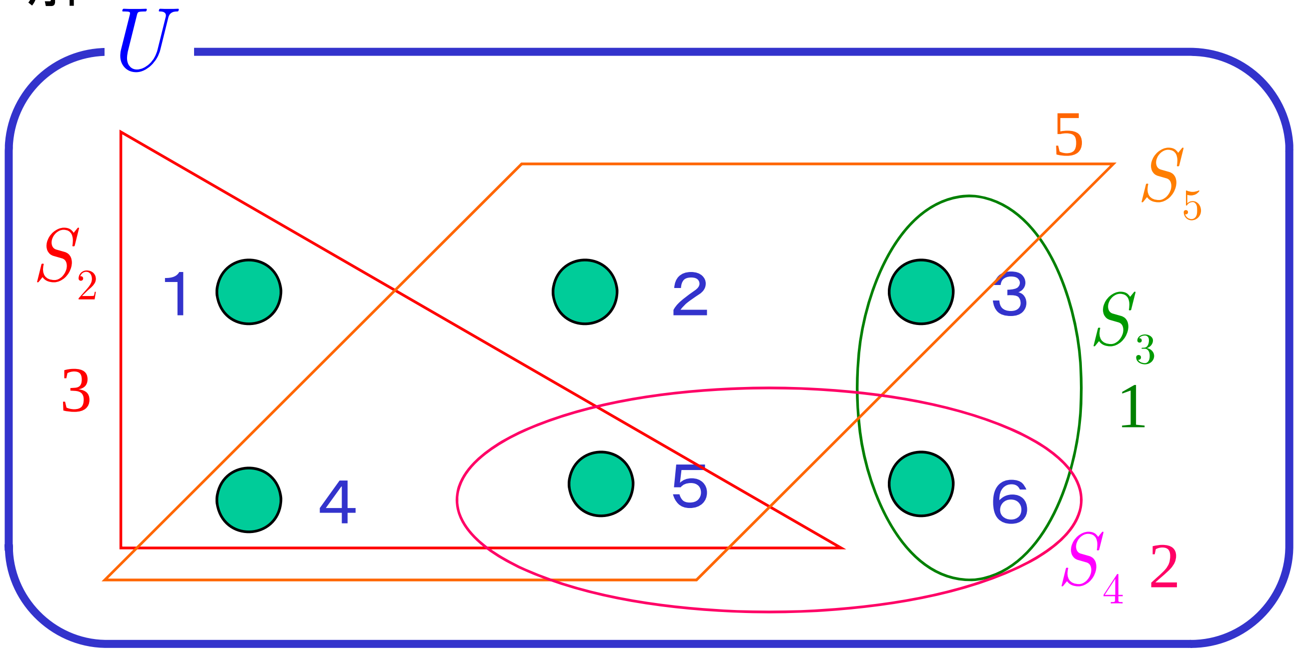
$\boldsymbol{x} = (0, 0, 1, 1, 1)$

$\boldsymbol{y} = (0, 2, 1, 0, 2, 0)$

$\boldsymbol{x} = (0, 1, 1, 1, 1)$

$\boldsymbol{y} = (0, 2, 1, 1, 2, 0)$

解



$$\boldsymbol{x} = (0, 1, 1, 1, 1)$$

$$\boldsymbol{y} = (0, 2, 1, 1, 2, 0)$$

$$f(\boldsymbol{x}) = 3 + 1 + 2 + 5 = 11$$

$$f^*(\boldsymbol{y}) = 3 \bullet (2 + 1 + 1 + 2) = 18$$

近似率

相補条件より、直ちに以下のように求められる。

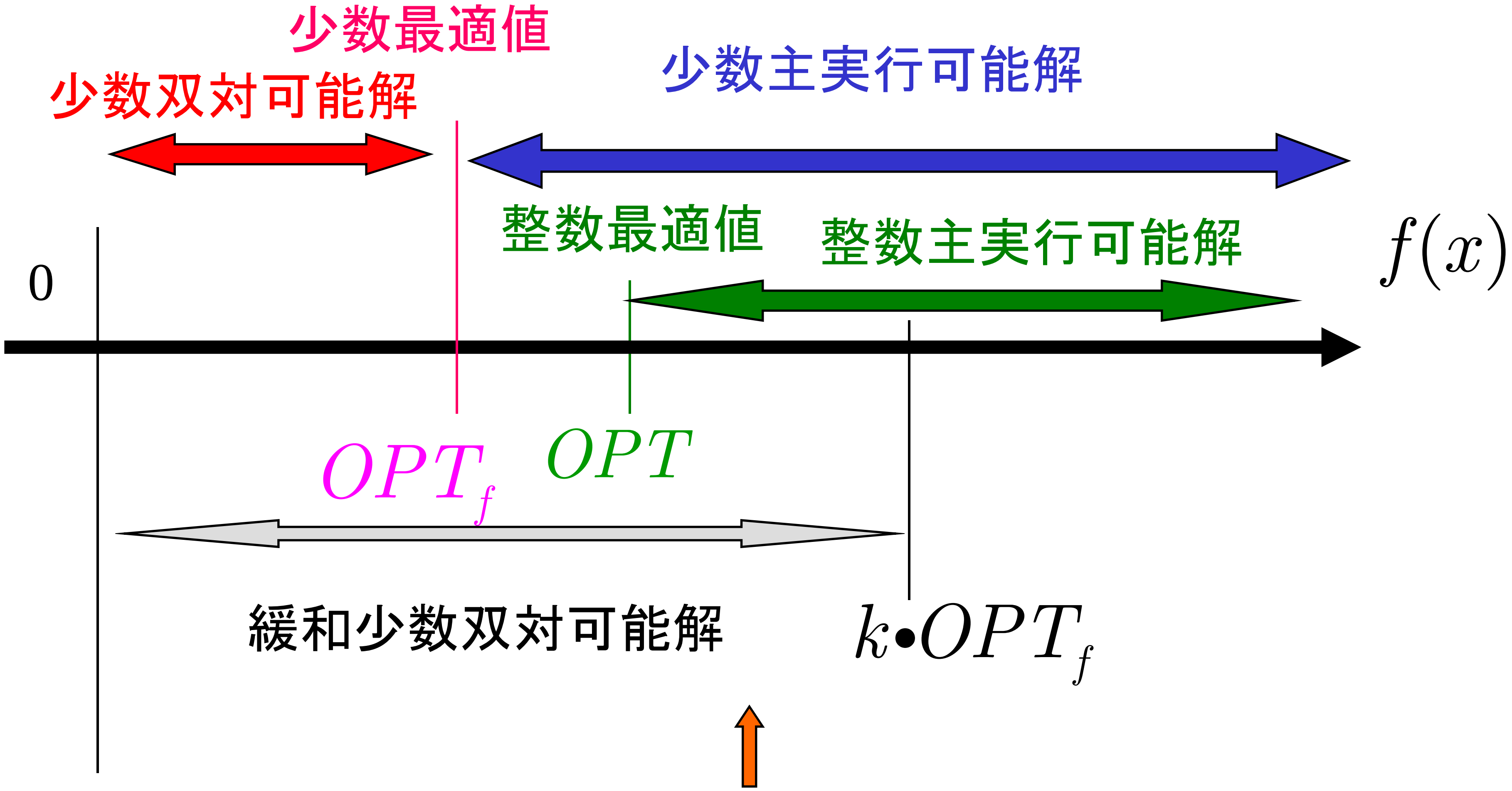
$$\begin{cases} \alpha = 1 \\ \beta = k(\text{頻度}) \end{cases}$$

$$\sum_{j=1}^{n} c_j x_j \leq (\alpha \bullet \beta) \sum_{i=1}^{m} b_i y_i$$

$$\therefore f^*(\boldsymbol{y}) \leq f(\boldsymbol{x}) \leq k f^*(\boldsymbol{y})$$

よって、k-近似アルゴリズムである。

関数値の関係



アルゴリズムでは、相補条件より
この範囲に含まれる整数解が得られる。

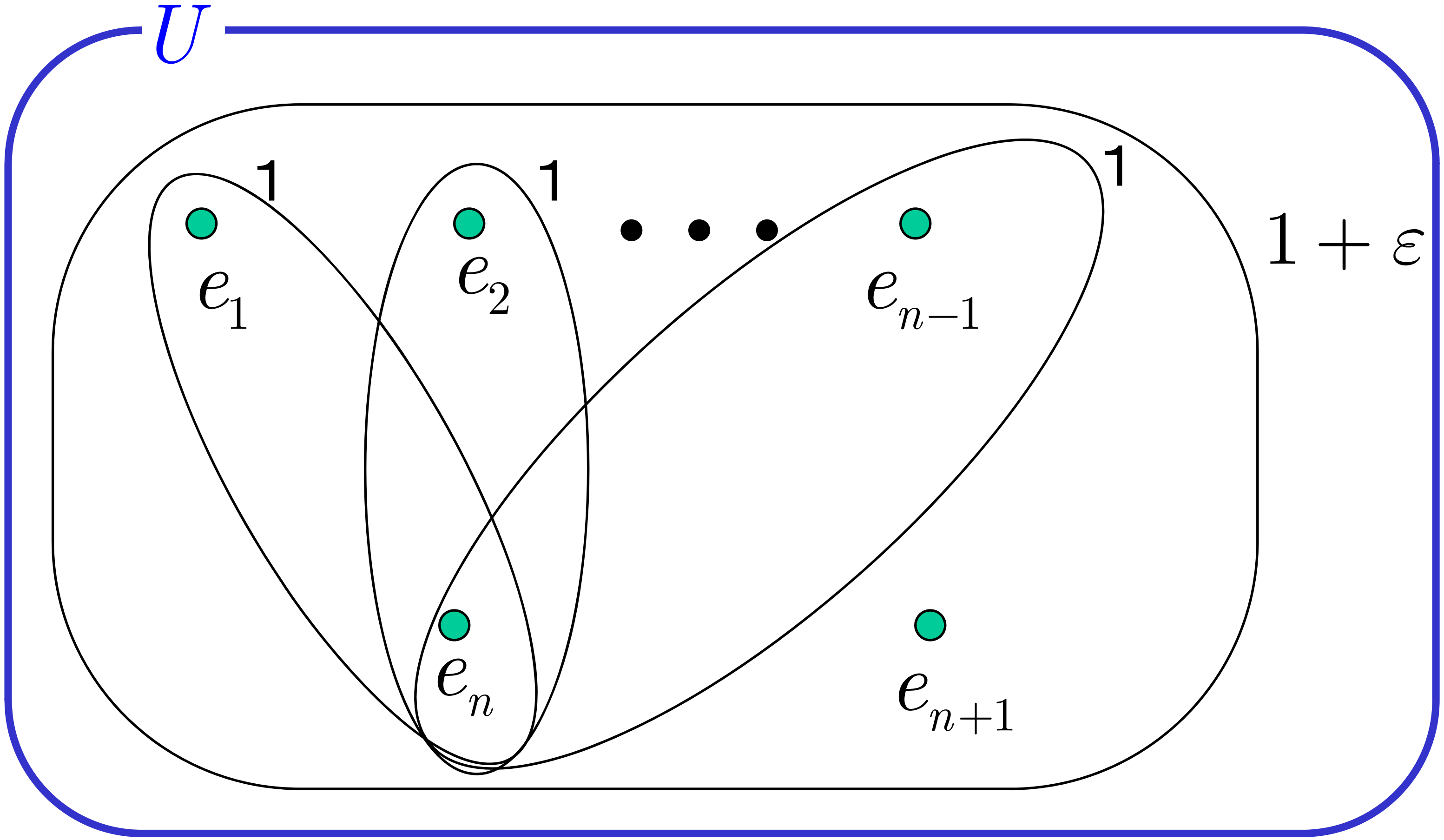
アルゴリズムの正当性

アルゴリズムでは、　　の更新は整数性を満たしている。
しかも、以下の2つを満足している。

〇すべての要素がカバーされている。
〇すべての集合が、オーバーパックされていない。
（タイトな集合を選んでいくので自動的に満足する。）

以上より、アルゴリズムは、集合カバーに対する
整数の実行可能解を出力する。

最悪の問題例



最適値 $1+\varepsilon$

アルゴリズムの出力 $n+\varepsilon$
（　　を最初に選ぶ）