離散最適化の数理 — 劣モジュラ構造のおもしろさ

藤重 悟

京都大学数理解析研究所

(京都大学附置研究所・センター品川セミナー，2012 年 3 月 2 日)

概 要

近年、「最適化」という言葉が広く使われるようになってきました。最適化の対象は人間の諸活動にかかわり、工学諸問題は勿論のこと、経済、経営、社会、環境、情報の諸問題など、あらゆる現実の実際的問題において、最適化の理論と技法が活用され、人類の福祉向上に役立てられています。また、今後さらに有効に使われるべき多くの実際的問題があると考えられます。

本講演では、個数などの (非負) 整数あるいは処理の順序や物の配置などの組合せ的な関係を表す離散量を扱う「離散最適化」の問題についてお話しします。そして、離散最適化問題の数理的な構造を、離散的な凸性としてとらえられる「劣モジュラ性」という切り口からながめて、その「おもしろさ」をお伝えしたいと思います。

最適化 (数理計画)

最適化 (数理計画)

オペレーションズ・リサーチ

最適化 (数理計画)

オペレーションズ・リサーチ＝“やりくり”学　(運籌学)

最適化 (数理計画)

オペレーションズ・リサーチ＝“やりくり”学　(運籌学)

「計画，設計，運用，経営，管理」の科学

最適化 (数理計画)

オペレーションズ・リサーチ＝**“やりくり”学**　(運籌学)

「計画，設計，運用，経営，管理」の科学

問題：制約条件，目的関数（評価関数）

(確定的・確率的，連続変数・離散変数，...)

最適化 (数理計画)

オペレーションズ・リサーチ＝**“やりくり”学**　(運籌学)

「計画，設計，運用，経営，管理」の科学

問題： 制約条件，目的関数（評価関数）

(確定的・確率的，連続変数・離散変数，...)

問題と最適化：

$$
\begin{array}{ccccc}
\text{モデル構築} & \Longrightarrow & \text{最適化} & \Longrightarrow & \text{モデルの評価} \\
\Uparrow & \Longleftarrow & \Longleftarrow & \Longleftarrow & \Downarrow
\end{array}
$$

最適化 (数理計画)

オペレーションズ・リサーチ＝**“やりくり”学**　(運籌学)

「計画，設計，運用，経営，管理」の科学

問題： 制約条件，目的関数（評価関数）

(確定的・確率的，連続変数・離散変数，...)

問題と最適化：

モデル構築 $\Longrightarrow$ 最適化 $\Longrightarrow$ モデルの評価

$\Uparrow$ $\Longleftarrow$ $\Longleftarrow$ $\Longleftarrow$ $\Downarrow$

連続最適化と離散最適化 (組合せ最適化)

離散最適化：整数値，配置，関係，(処理) 順序，組合せ，. . .
(組合せ最適化)

離散最適化，組合せ最適化

(例) 最短経路 (カーナビ)，配送計画

スケジューリング (順序，組み立て，人員配置，日程計画，...)

生産計画・工程管理，SCM，ロジスティクス，物流

(通信，交通) ネットワーク設計，ネットワーク連結度・信頼性

VLSI 設計，画像処理，CAD

建築構造物設計，施設配置，避難誘導，都市計画

バイオインフォマティクス・計算生物学，データマイニング

非分割財の経済，資源配分，オークション

⋮

離散最適化，組合せ最適化

(例) 最短経路 (カーナビ)，配送計画

スケジューリング (順序，組み立て，人員配置，日程計画，...)

生産計画・工程管理，SCM，ロジスティクス，物流

(通信，交通) ネットワーク設計，ネットワーク連結度・信頼性

VLSI 設計，画像処理，CAD

建築構造物設計，施設配置，避難誘導，都市計画

バイオインフォマティクス・計算生物学，データマイニング

非分割財の経済，資源配分，オークション

⋮

“有限個” の可能な解から最適な解を見つける！

離散最適化，組合せ最適化

(例)　最短経路 (カーナビ)，配送計画

スケジューリング (順序，組み立て，人員配置，日程計画，...)

生産計画・工程管理，SCM，ロジスティクス，物流

(通信，交通) ネットワーク設計，ネットワーク連結度・信頼性

VLSI 設計，画像処理，CAD

建築構造物設計，施設配置，避難誘導，都市計画

バイオインフォマティクス・計算生物学，データマイニング

非分割財の経済，資源配分，オークション

⋮

“有限個” の可能な解から最適な解を見つける！

“有限個” ＝途轍もなく大きな数

“有限個” の可能な解から最適な解を見つける！

“有限個” ＝途轍もなく大きな数

(例) １個調べるのに 10^{-6} 秒かかるときに必要な総時間

集合の大きさ n	20	50	100
部分集合全体 2^n			

“有限個” の可能な解から最適な解を見つける！

“有限個” ＝途轍もなく大きな数

(例) 1個調べるのに 10^{-6} 秒かかるときに必要な総時間

集合の大きさ n	20	50	100
部分集合全体 2^n	1秒	35年	3×10^6年

“有限個” の可能な解から最適な解を見つける！

“有限個” ＝途轍もなく大きな数

(例) 1 個調べるのに 10^{-6} 秒かかるときに必要な総時間

集合の大きさ n	20	50	100
部分集合全体 2^n	1 秒	35 年	3×10^6年
$10n^3$			
$1000n$			

“有限個” の可能な解から最適な解を見つける！

“有限個” ＝途轍もなく大きな数

(例) １個調べるのに 10^{-6} 秒かかるときに必要な総時間

集合の大きさ n	20	50	100
部分集合全体 2^n	1 秒	35 年	3×10^6 年
$10n^3$	0.08 秒	1 秒	10 秒
$1000n$	0.02 秒	0.05 秒	0.1 秒

効率の良いアルゴリズムの研究の重要性

問題解決の鍵は？

問題解決の鍵は

与えられた問題の持つ有効な数理的構造の解明！

問題解決の鍵は

与えられた問題の持つ有効な数理的構造の解明！

⇓

劣モジュラ構造　(マトロイド構造)

(効率的なアルゴリズムに結び付く離散構造)

E: 有限集合

集合関数 $f : 2^E \to \mathbf{R}$

$X \subseteq E \ \mapsto \ f(X) \in \mathbf{R}$

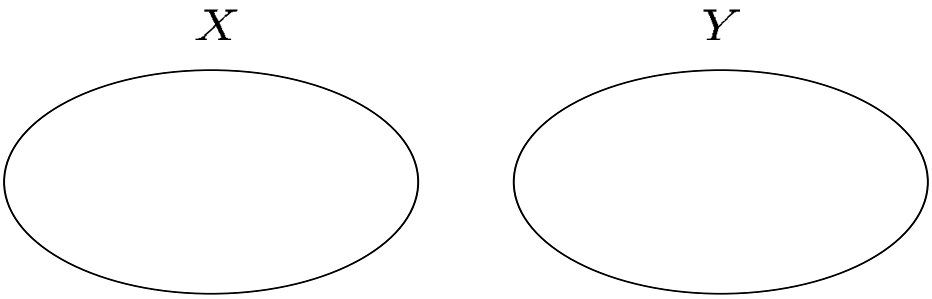
(以下では，$f(\emptyset) = 0$ と仮定する.)

E: 有限集合

集合関数 $f : 2^E \to \mathbf{R}$

＜劣加法性＞

$$f(X) + f(Y) \geq f(X \cup Y) \qquad (\forall X,\, Y \subseteq E,\ X \cap Y = \emptyset)$$



E: 有限集合

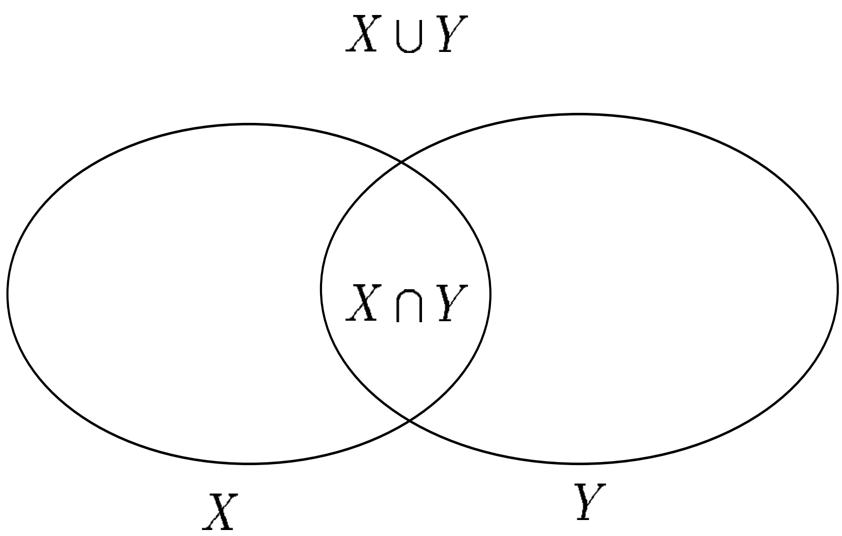
集合関数 $f : 2^E \to \mathbf{R}$

＜劣加法性＞

$$f(X) + f(Y) \geq f(X \cup Y) \qquad (\forall X,\, Y \subseteq E,\, X \cap Y = \emptyset)$$

＜劣モジュラ性＞

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \qquad (\forall X,\, Y \subseteq E)$$



E: 有限集合

集合関数 $f : 2^E \to \mathbf{R}$

＜劣加法性＞

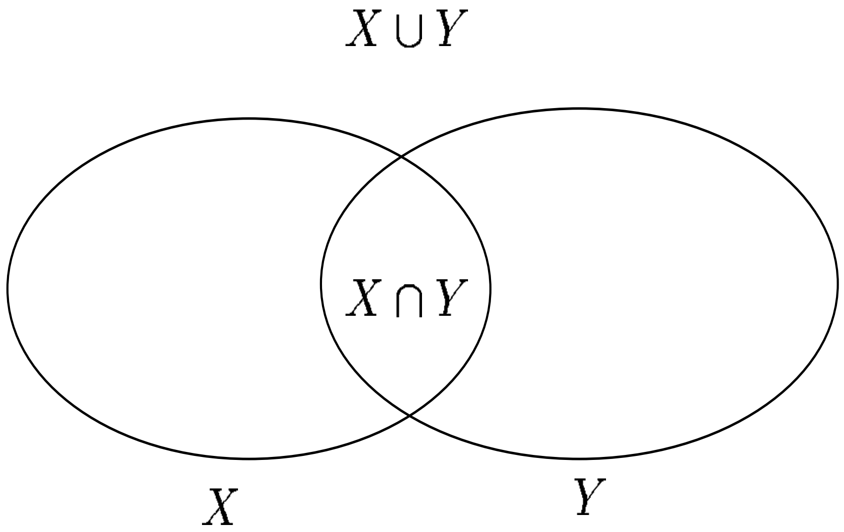
$$f(X) + f(Y) \geq f(X \cup Y) \qquad (\forall X,\, Y \subseteq E,\, X \cap Y = \emptyset)$$

＜劣モジュラ性＞

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \qquad (\forall X,\, Y \subseteq E)$$

(1) $\{f(X) - f(X \cap Y)\} + \{f(Y) - f(X \cap Y)\}$
$\geq f(X \cup Y) - f(X \cap Y)$

(2) $f(X) - f(X \cap Y) \geq f(X \cup Y) - f(Y)$



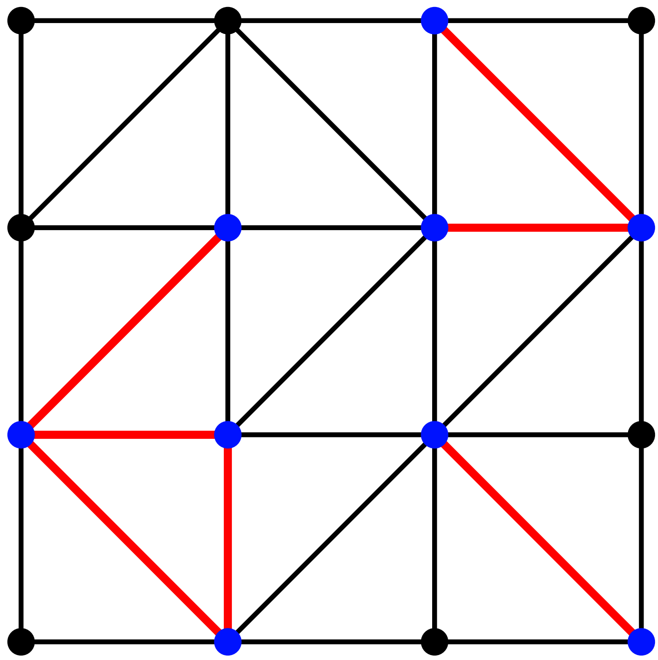
(注) $X \setminus Y$ と $Y \setminus X$ の各要素の数を 1 に限っても同値.

劣モジュラ関数の例:

$G=(V,E)$: **点集合** V と**枝集合** E をもつ**グラフ**

グラフ $G=(V,E)$ **の階数関数** $f: 2^E \to \mathbf{Z}_+$

$\forall X \subseteq E: f(X) = (X$ で張られる点数$) - ($連結成分の個数$)$

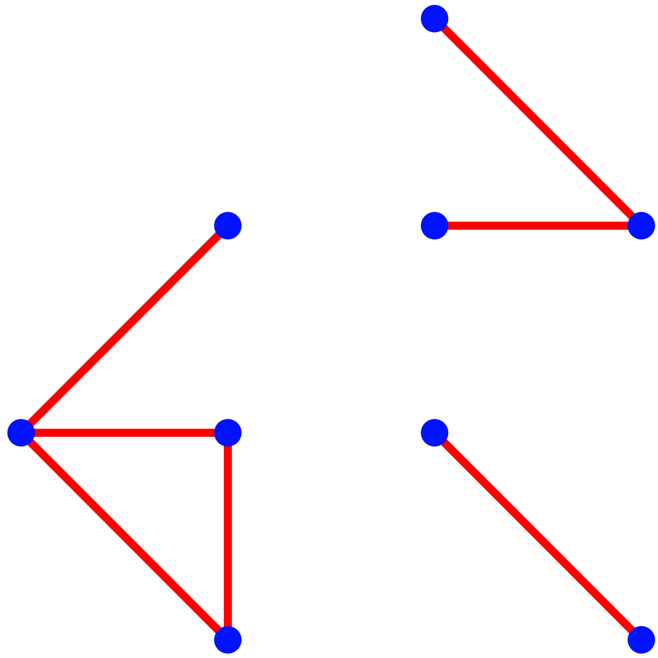


劣モジュラ関数の例:

$G=(V,E)$: **点集合** V と**枝集合** E をもつ**グラフ**

グラフ $G=(V,E)$ **の階数関数** $f: 2^E \to \mathbf{Z}_+$

$\forall X \subseteq E: f(X) = (X$ で張られる点数$) - ($連結成分の個数$)$



$f(X) = 9 - 3 = 6$

(注) f は，単調非減少な劣モジュラ関数である．

行列の階数関数

M: $m \times n$ **行列** (matrix)

$E = \{1, 2, \cdots, n\}$: M の**列の集合**

$$
\begin{array}{cc}
 & \begin{array}{cccc} 1 & 2 & \cdots & n \end{array} \\
M = & \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}
\end{array}
$$

$\forall X \subseteq E$: $f(X) = M$ の X に対応する列からなる**部分行列の階数**

(注) f は，単調非減少な劣モジュラ関数である．

行列の階数関数

M: $m \times n$ 行列 (matrix)

$E = \{1, 2, \cdots, n\}$: M の列の集合

$$M = \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & n \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \end{array}$$

$\forall X \subseteq E:\ f(X) = M$ の X に対応する列からなる部分行列の階数

$$\mathcal{I} = \{X \mid X \subseteq E,\ f(X) = |X|\} \qquad (|X|: X \text{ の要素数 (列数)})$$

(1) $\emptyset \in \mathcal{I}$.

(2) $I \in \mathcal{I}$ かつ $J \subset I$ ならば, $J \in \mathcal{I}$.

(3) $I,\ J \in \mathcal{I}$ かつ $|I| < |J|$ ならば，ある要素 $e \in J \setminus I$ が存在して，$I \cup \{e\} \in \mathcal{I}$.

(対 $(E, \mathcal{I})$ は，**マトロイド** (matroid)，$\mathcal{I}$ はその**独立集合族**と呼ばれる.)

(マトロイド：双対性，豊富な演算と変換，効率的なアルゴリズム)

グラフの場合，グラフの階数関数 f に対して,

$$\mathcal{I} = \{X \mid X \subseteq E,\ f(X) = |X|\}$$

(1) $\emptyset \in \mathcal{I}$.

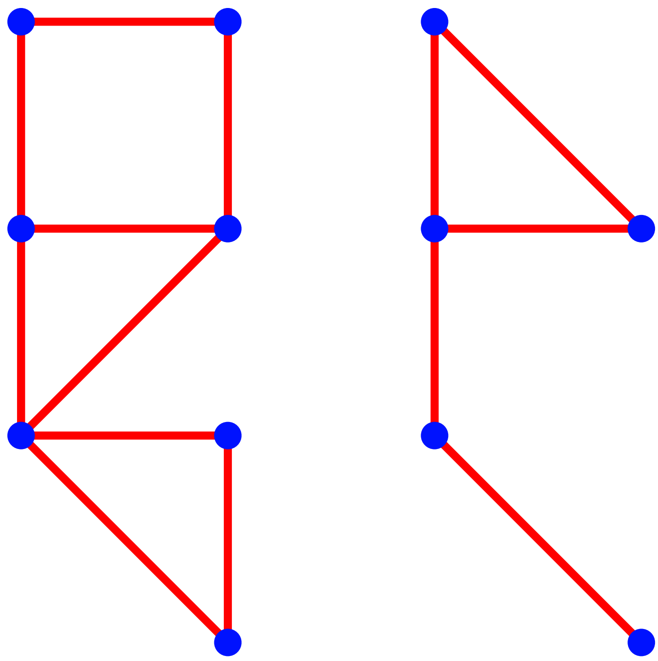
(2) $I \in \mathcal{I}$ かつ $J \subset I$ ならば, $J \in \mathcal{I}$.

(3) $I,\ J \in \mathcal{I}$ かつ $|I| < |J|$ ならば，ある要素 $e \in J \setminus I$ が存在して, $I \cup \{e\} \in \mathcal{I}$.

の各独立集合 $I \in \mathcal{I}$ は,「閉路を含まない枝集合」に対応する.

逆に,

$$f(X) = \max\{|I| \mid I \subseteq X,\ \ I \in \mathcal{I}\} \qquad (\forall X \subseteq E)$$



グラフの場合，グラフの階数関数 f に対して，

$$\mathcal{I} = \{X \mid X \subseteq E,\ f(X) = |X|\}$$

(1) $\emptyset \in \mathcal{I}$.

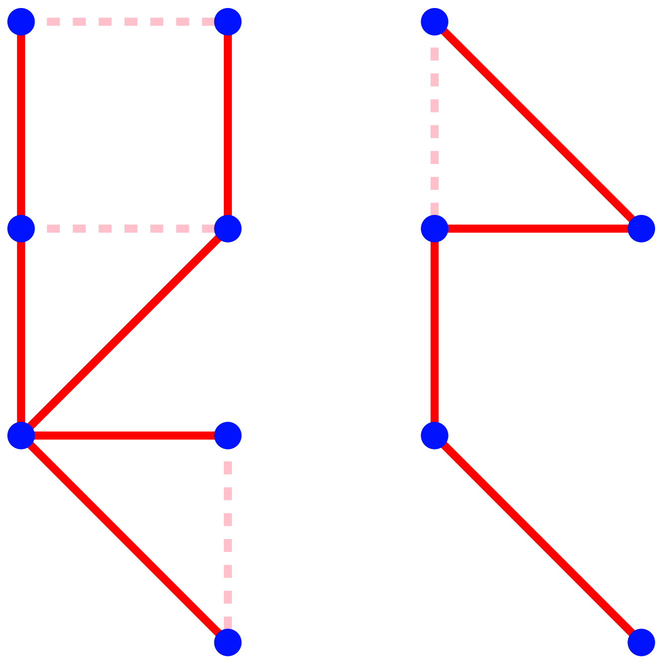
(2) $I \in \mathcal{I}$ かつ $J \subset I$ ならば, $J \in \mathcal{I}$.

(3) $I,\ J \in \mathcal{I}$ かつ $|I| < |J|$ ならば，ある要素 $e \in J \setminus I$ が存在して，$I \cup \{e\} \in \mathcal{I}$.

の各独立集合 $I \in \mathcal{I}$ は,「閉路を含まない枝集合」に対応する.

逆に,

$$f(X) = \max\{|I| \mid I \subseteq X,\ \ I \in \mathcal{I}\} \qquad (\forall X \subseteq E)$$

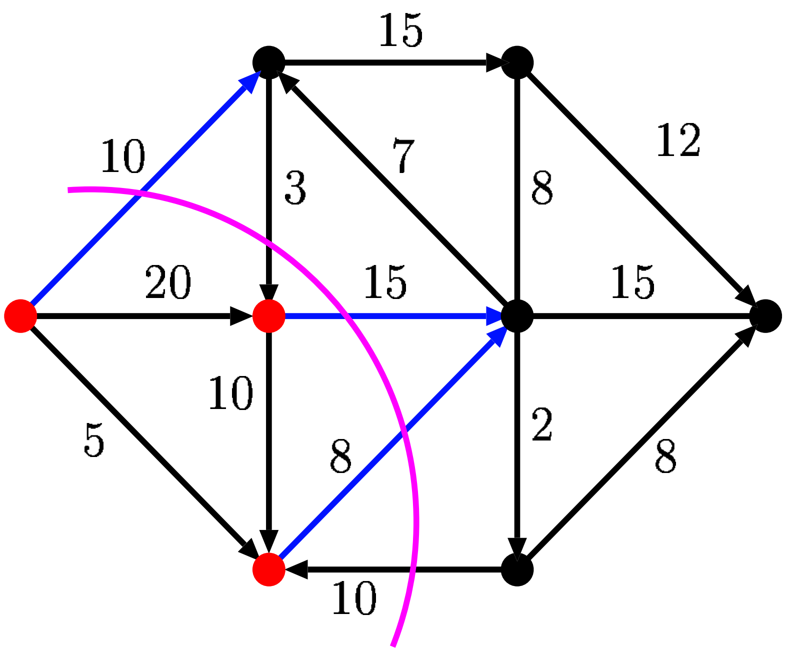


ネットワークのカット関数

$G = (V, A)$: **(有向) グラフ**

$c(a)$: **枝** $a \in A$ **の容量** ≥ 0

$\forall X \subseteq V$: $f(X) = X$ の中から**出る枝の容量の総和**



$$f(X) = 33$$

(注) f は，必ずしも単調非減少でない劣モジュラ関数.

多端子ネットワークの流量関数

入り口１個で，**複数の出口**があるネットワークにおいて，

T: 出口全体の集合

$\forall X \subseteq T$: $f(X) =$ **出口** X への**流出量総和の最大値**

(注) f は，単調非減少な劣モジュラ関数である．

エントロピー関数

$x_1, x_2, \cdots, x_n$: 1 から N の整数値をとる確率変数

$E = \{x_1, x_2, \cdots, x_n\}$

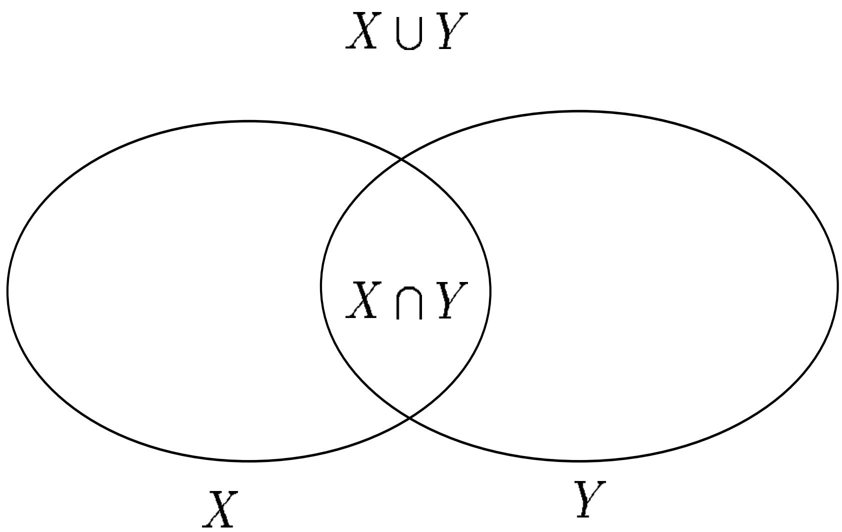
$\forall X \subseteq E$: $f(X) =$ (X の (シャノンの意味での) エントロピー)

たとえば $X = \{x_1, x_2, \cdots, x_k\}$ $(1 \leq k \leq n)$ であるとき,

$$f(X) = -\sum_{i_1=1}^{N} \cdots \sum_{i_k=1}^{N} P(x_1{=}i_1, \cdots, x_k{=}i_k) \log_2 P(x_1{=}i_1, \cdots, x_k{=}i_k)$$

(注) f は単調非減少な劣モジュラ関数である.

$$f(X) - f(X \cap Y) \geq f(X \cup Y) - f(Y)$$



凸ゲーム

E: プレイヤーの集合

$\forall X \subseteq E$: $f(X) =$ (X が協力する場合に X 全体でかかる費用)

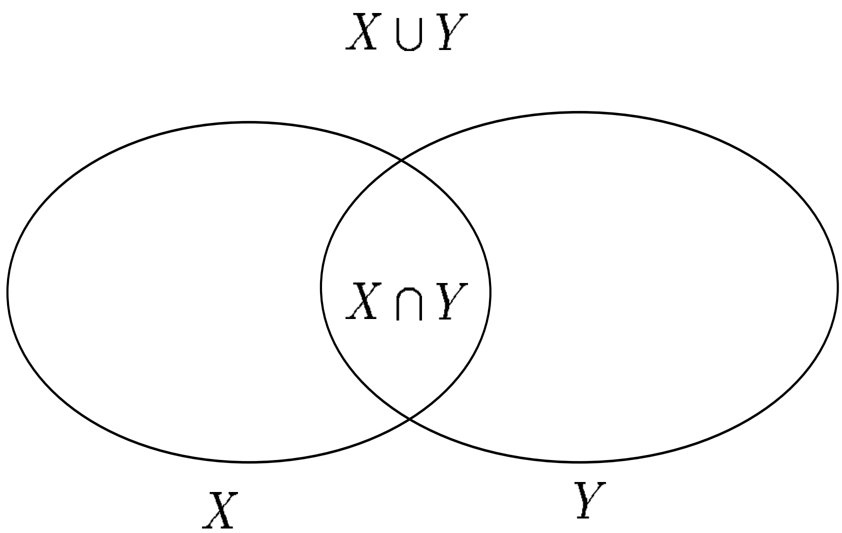
f が**単調非減少な劣モジュラ関数**であるとき，**凸ゲーム**という.

$x(e)$: プレイヤー e の分担金 (≥ 0)

$$\forall X \subset E: \ x(X) \leq f(X), \qquad x(E) = f(E)$$

$$\left(x(X) = \sum_{e \in X} x(e) \right)$$

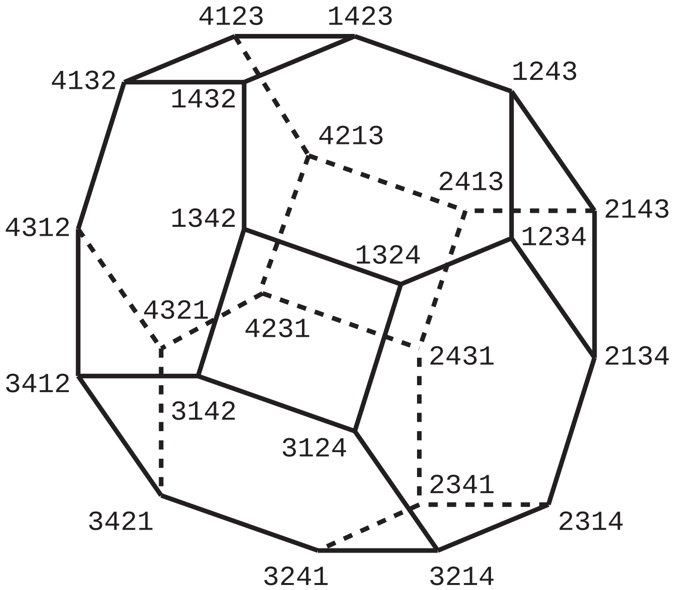
$$f(X) - f(X \cap Y) \geq f(X \cup Y) - f(Y)$$



置換多面体 (permutahedron)

$E = \{1, 2, \cdots, n\}$

E の**置換** $\sigma : E \to E$ による $\mathbf{R}^E$ 中の点 $(\sigma(1), \sigma(2), \cdots, \sigma(n))$ の集合の凸包を**置換多面体** P_n.



$$g(k) = \sum_{i=1}^{k}(n - i + 1) \qquad (k = 1, 2, \cdots, n)$$

$$f(X) = g(|X|) \qquad (X \subseteq E)$$

$$P_n = \{x \in \mathbf{R}^E \mid \forall X \subset E :\ x(X) \leq f(X),\ x(E) = f(E)\}$$

(注) f は単調増加な劣モジュラ関数である.

最大木問題（最小木問題)

(最大重み全域木問題)

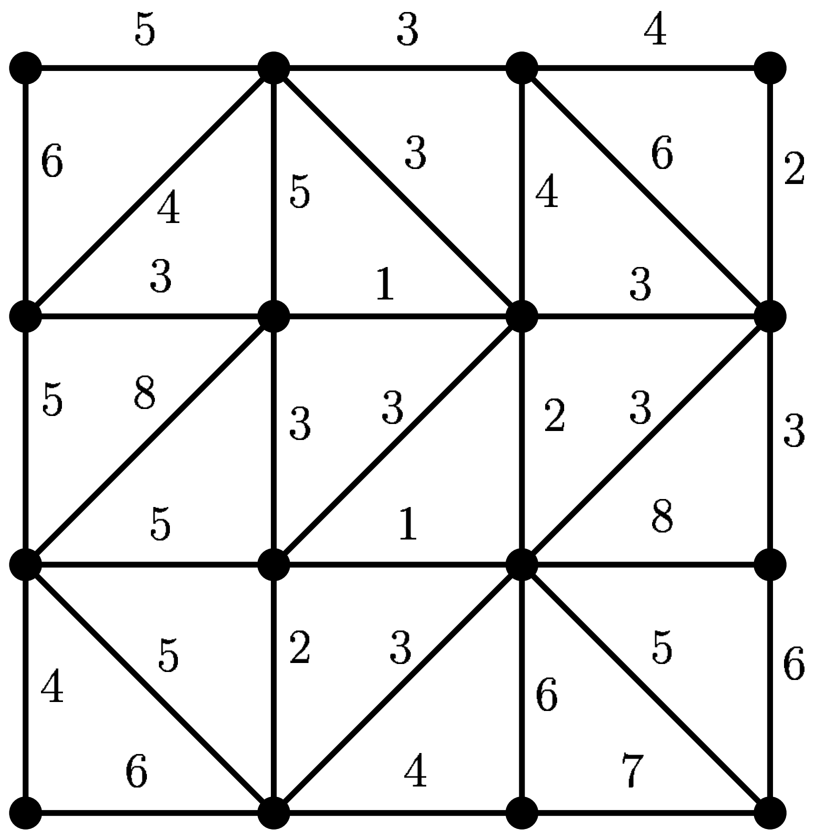
全域木＝閉路を含まない極大な枝集合

最大木問題

連結な**グラフ** $G=(V,E)$ と**枝重み関数** $w : E \to \mathbf{R}$

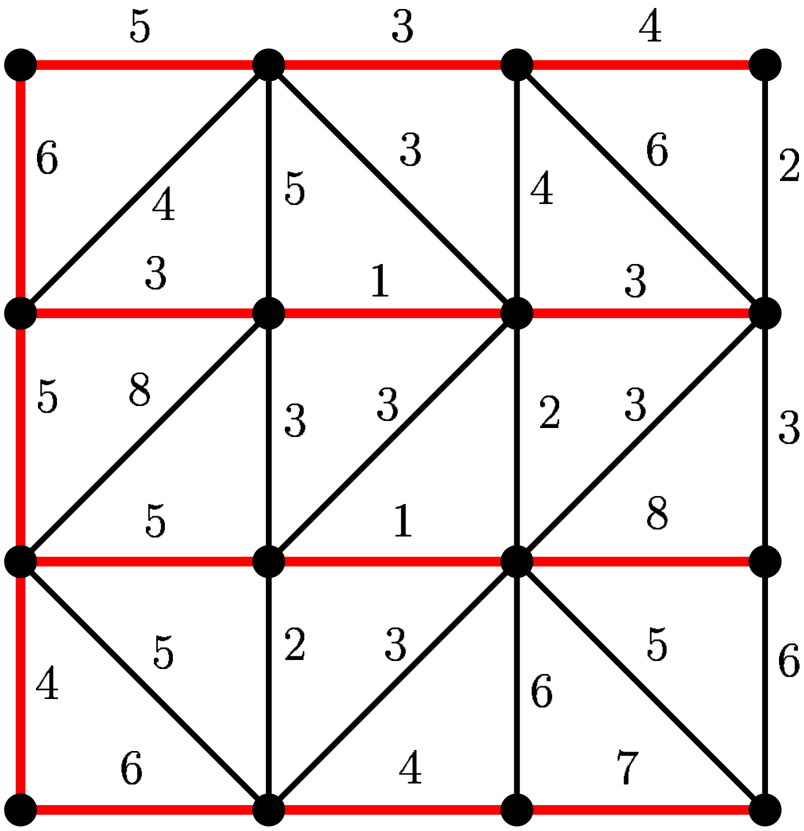
$$w(T) = \sum_{e \in T} w(e) \qquad (T \subseteq E\text{: } G \text{ の (全域) 木})$$

問題: **重み最大の (全域) 木を見出せ.**



最大木問題

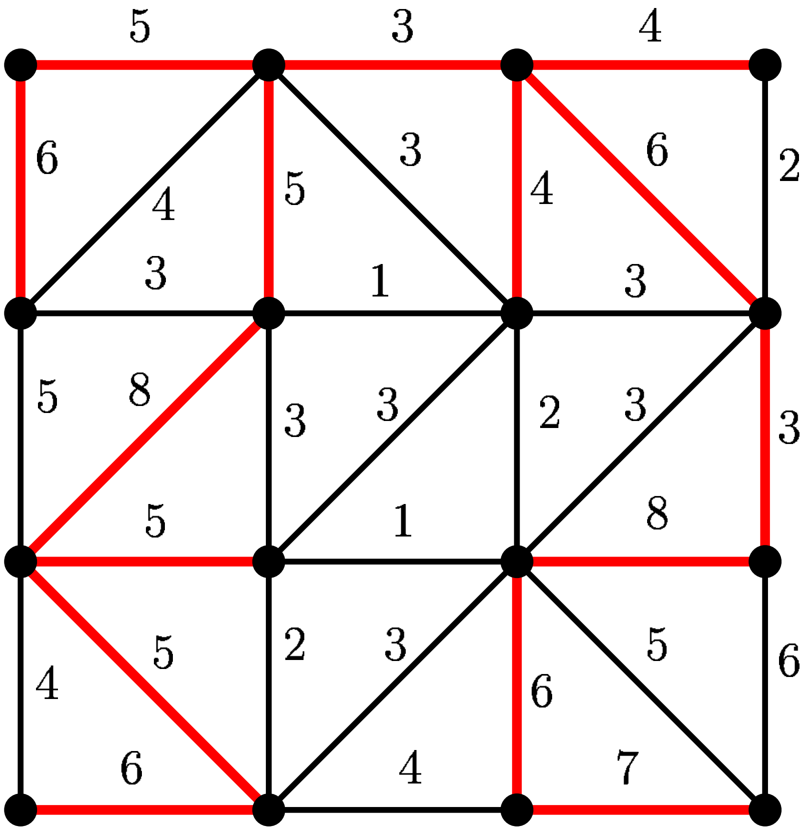
連結な**グラフ** $G = (V, E)$ と**枝重み関数** $w : E \to \mathbf{R}$



重みの総和 $w(T) = 65$ **最適か？**

最大木問題

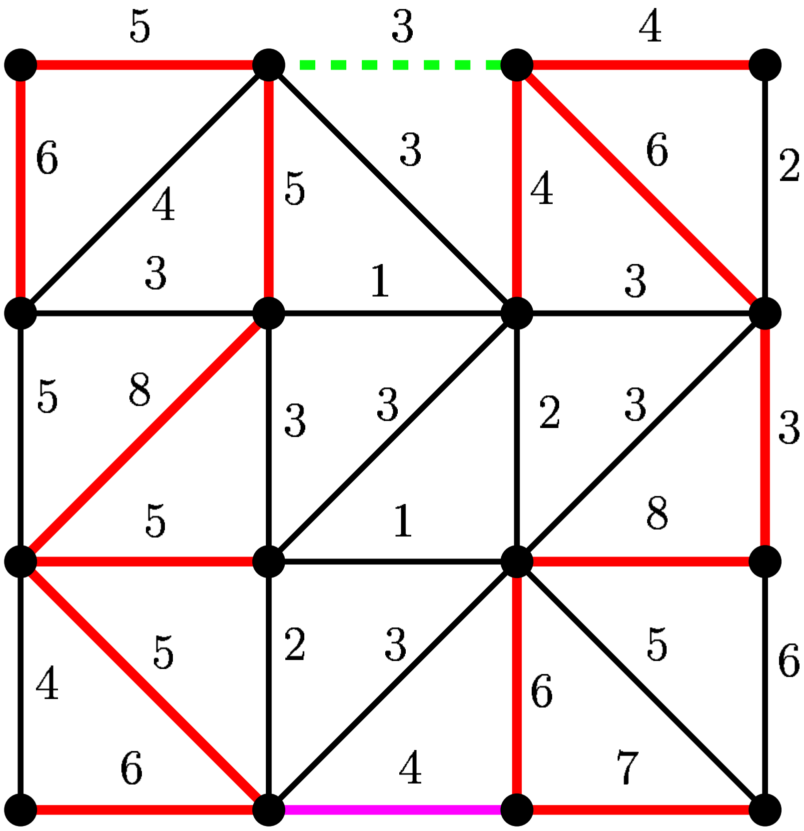
連結な**グラフ** $G = (V, E)$ と**枝重み関数** $w : E \to \mathbf{R}$



重みの総和 $w(T) = 81$　　**最適か？**

最大木問題

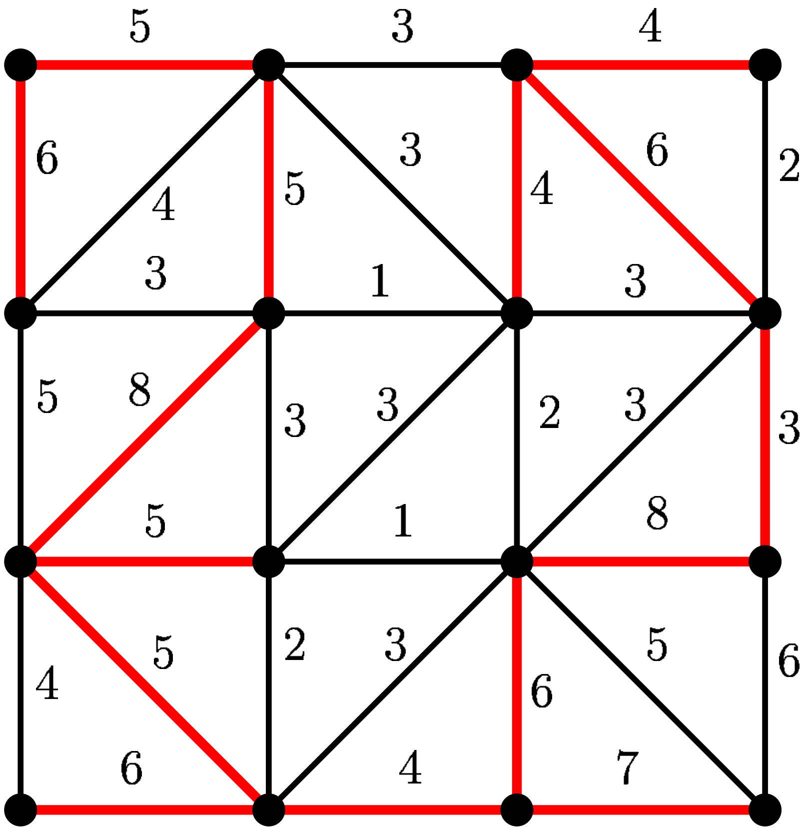
連結な**グラフ** $G = (V, E)$ と**枝重み関数** $w : E \to \mathbf{R}$



重みの総和 $w(T) = 82$　　**最適か？**

最大木問題

連結な**グラフ** $G = (V, E)$ と**枝重み関数** $w : E \to \mathbf{R}$



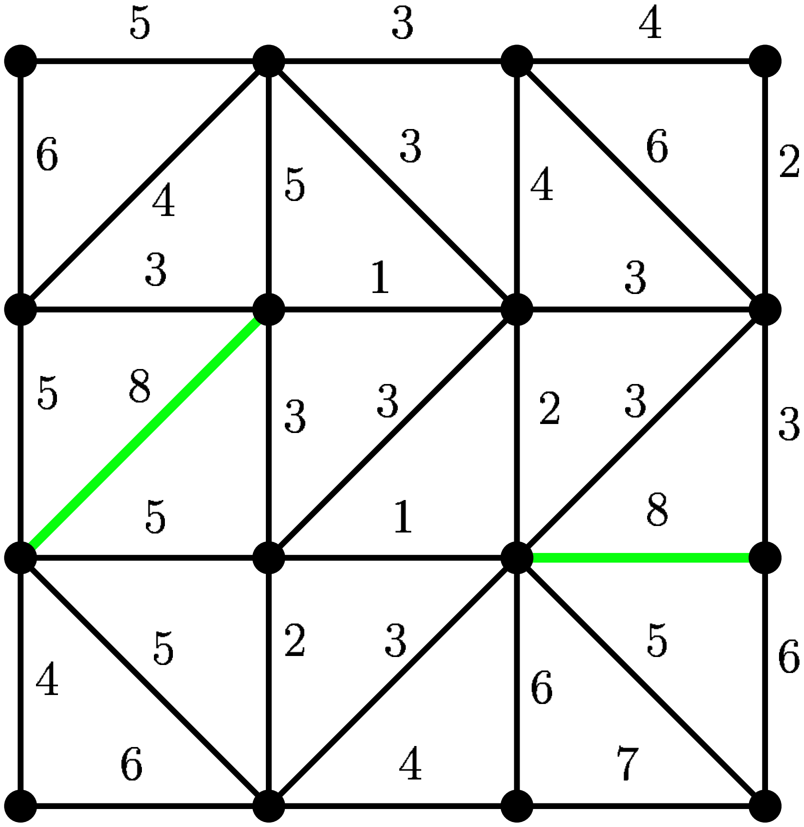
重みの総和 $w(T) = 82$ **最適である！**

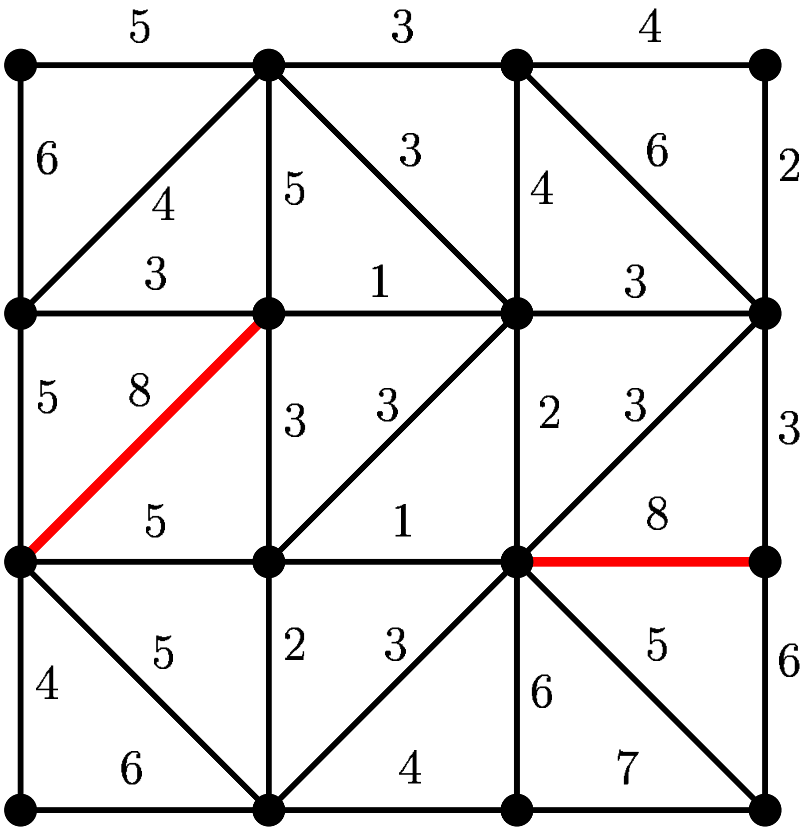
貪欲アルゴリズム (greedy algorithm)

Step 0: 重みの大きい順に並べて，$w(e_1) \geq w(e_2) \geq \cdots \geq w(e_n)$.
$T \leftarrow \emptyset$ とおく.

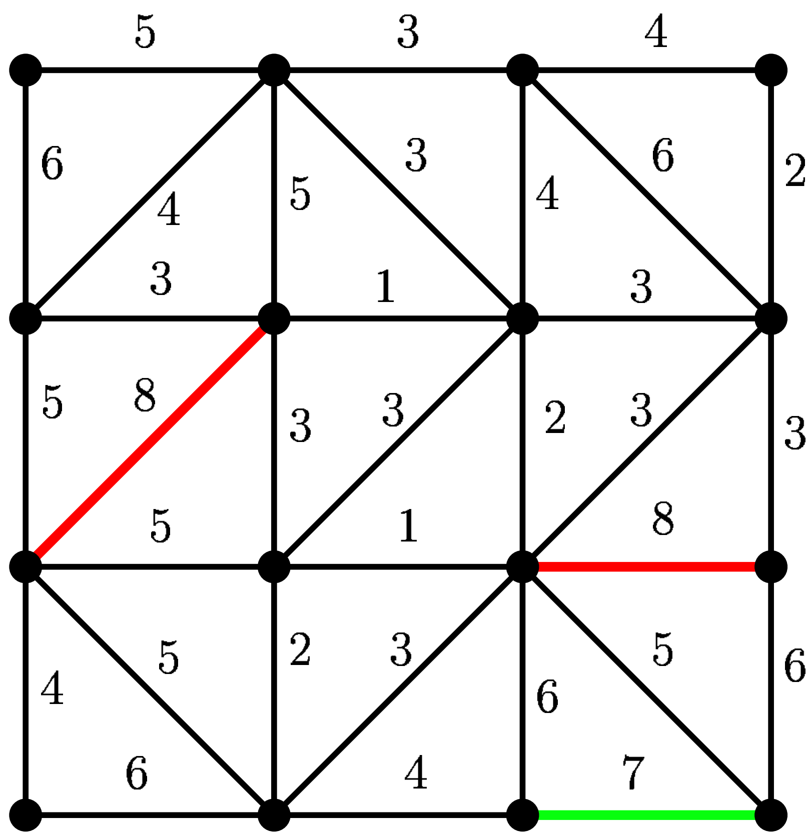
Step 1: $i = 1, 2, \cdots, n$ に対して,
　$T \cup \{e_i\}$ が閉路を含まなければ $T \leftarrow T \cup \{e_i\}$,
　T が (全域) 木ならば T を出力して停止.

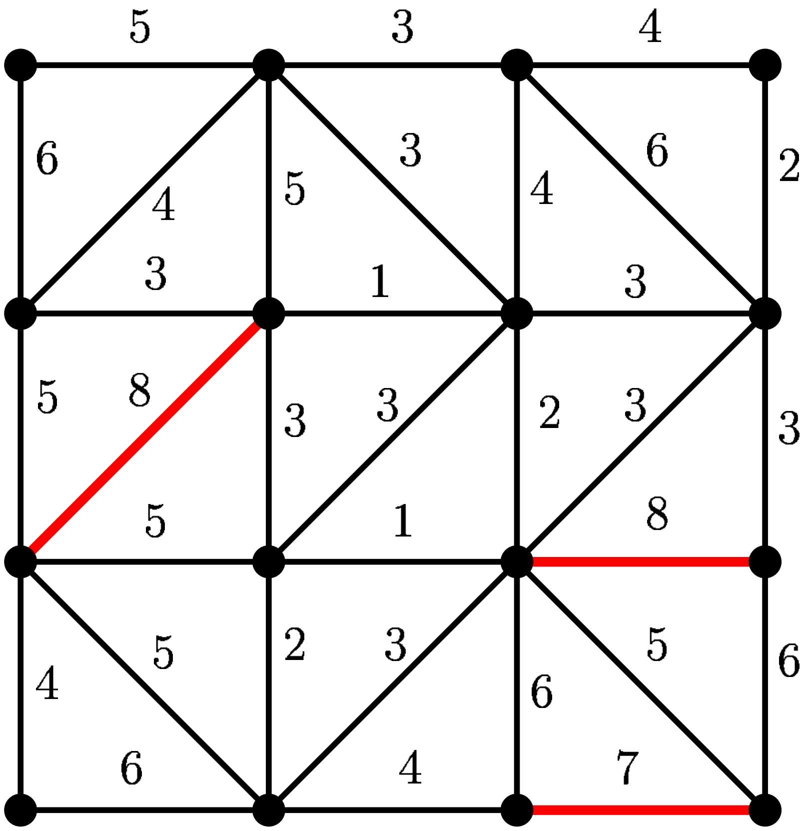
定理：貪欲アルゴリズムによって、最大重みの (全域) 木が求められる. また，任意の最大重みの (全域) 木は，ある単調非増加な重みの並べ方に対する貪欲アルゴリズムの解である. □

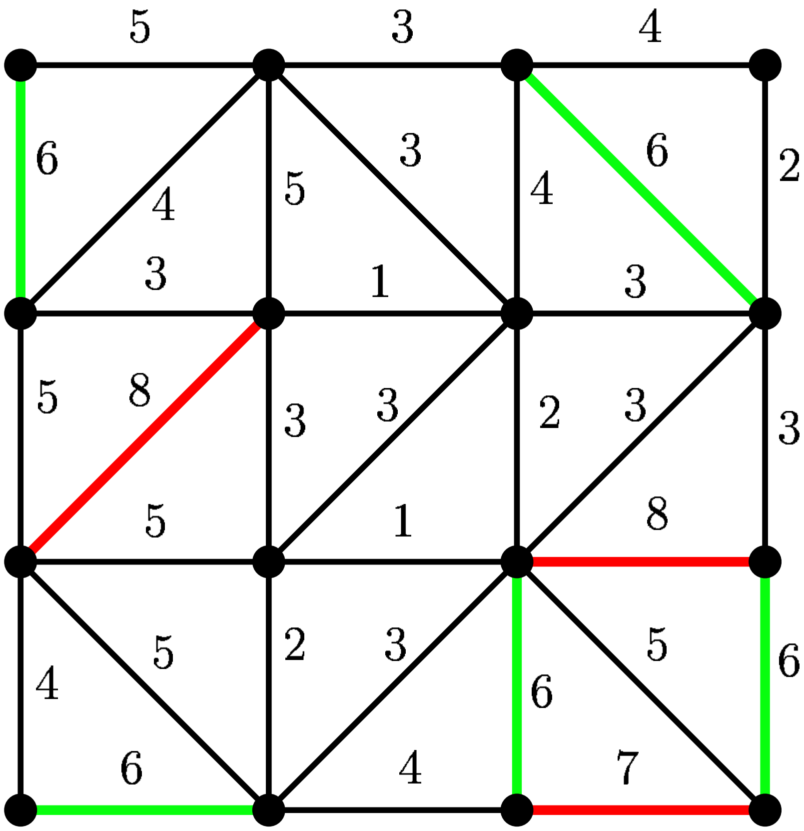

5
3
4
6
3
6
2
4
5
4
3
1
3
8
5
3
3
2
3
3
5
1
8
5
2
3
5
6
6
4
6
4
7

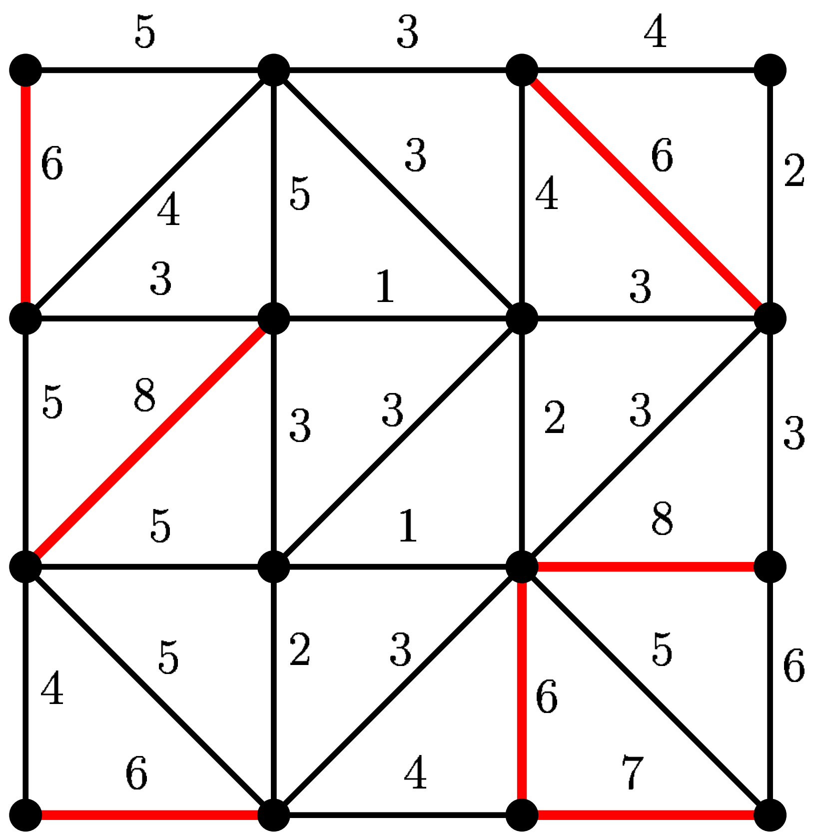


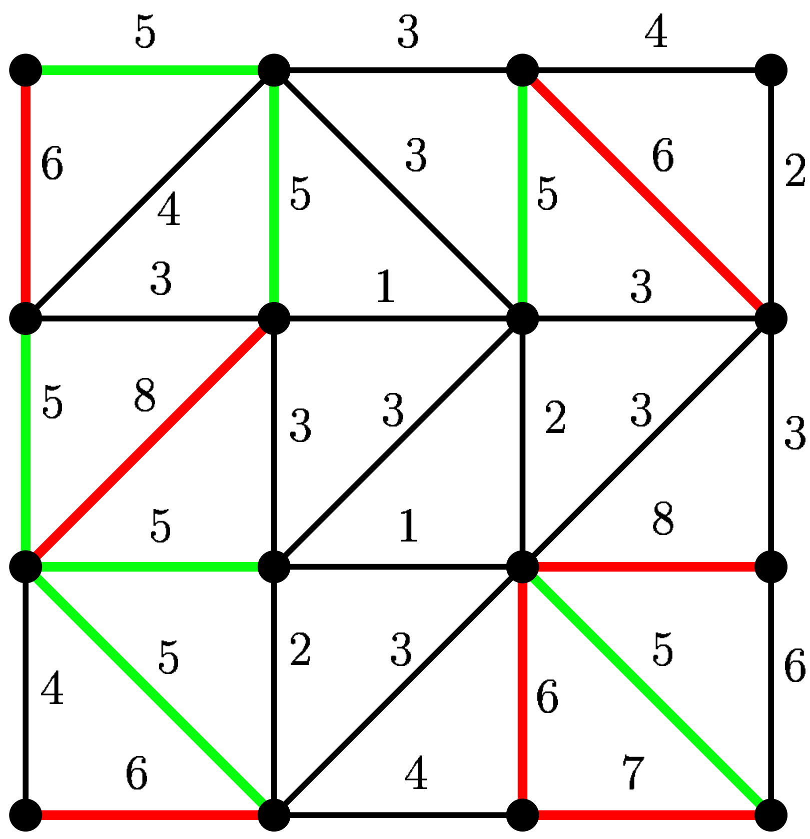
5
3
4
6
4
5
3
4
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7

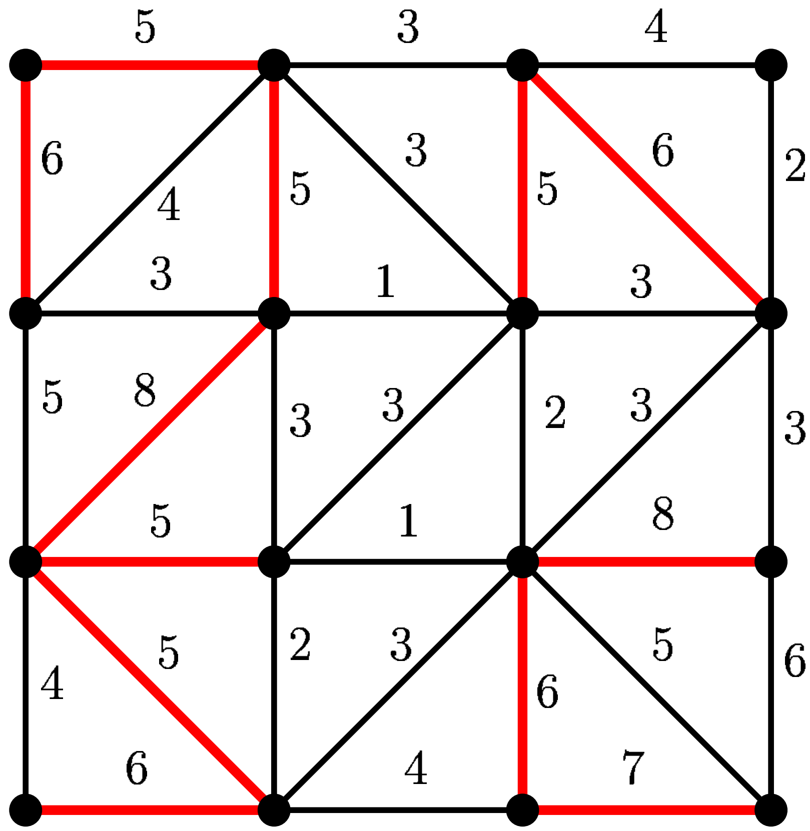

5
3
4
6
4
5
3
4
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7


5
3
4
6
4
5
3
4
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7

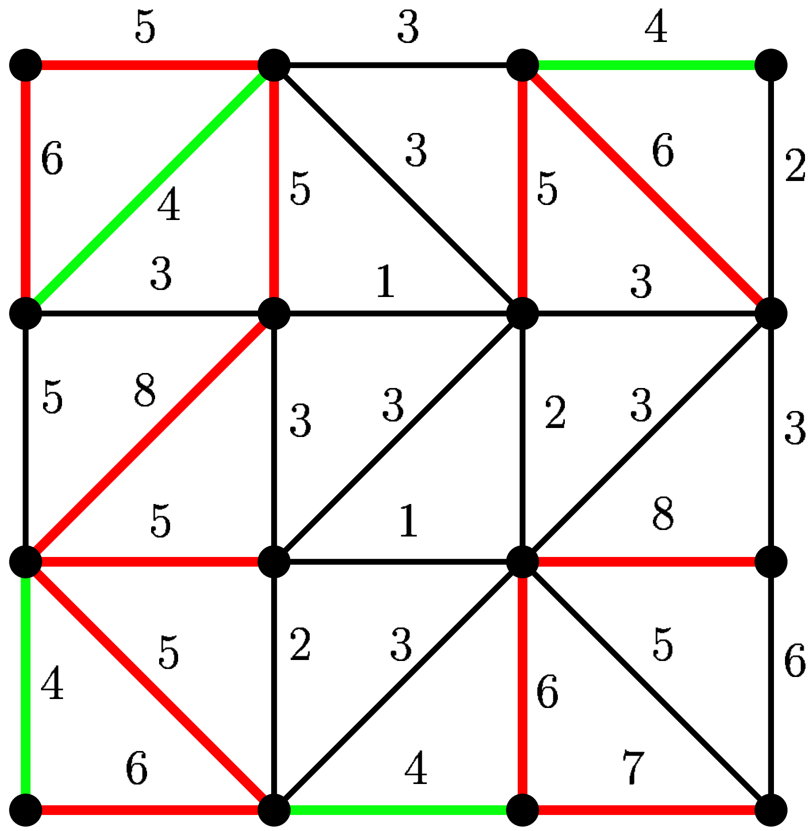

5
3
4
6
4
5
3
4
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7


5
3
4
6
4
5
3
4
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7

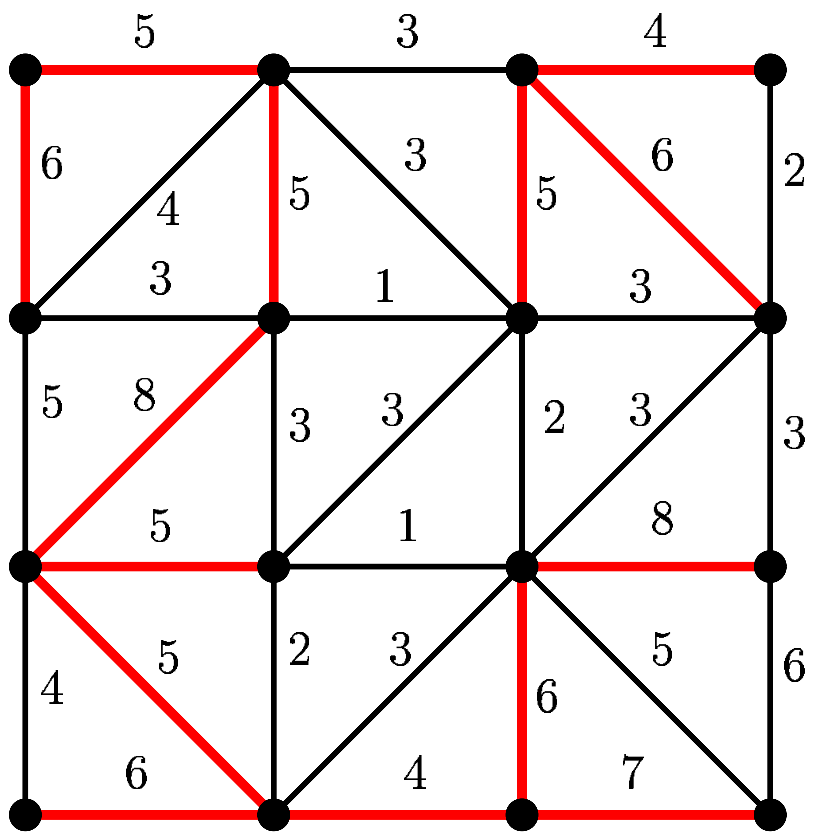

5
3
4
6
4
5
3
6
5
2
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7

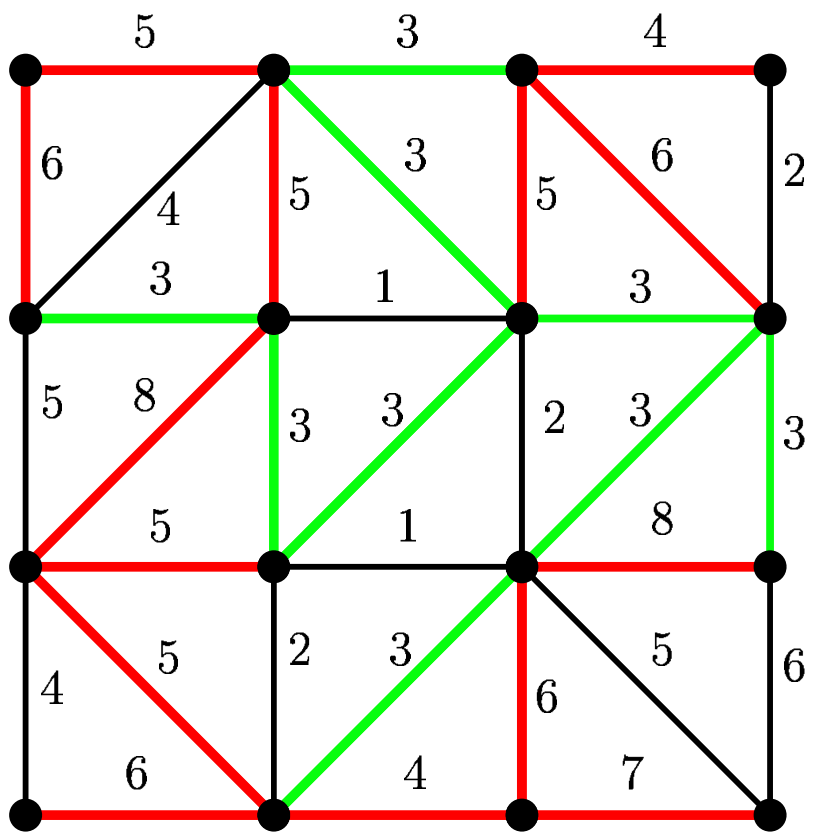


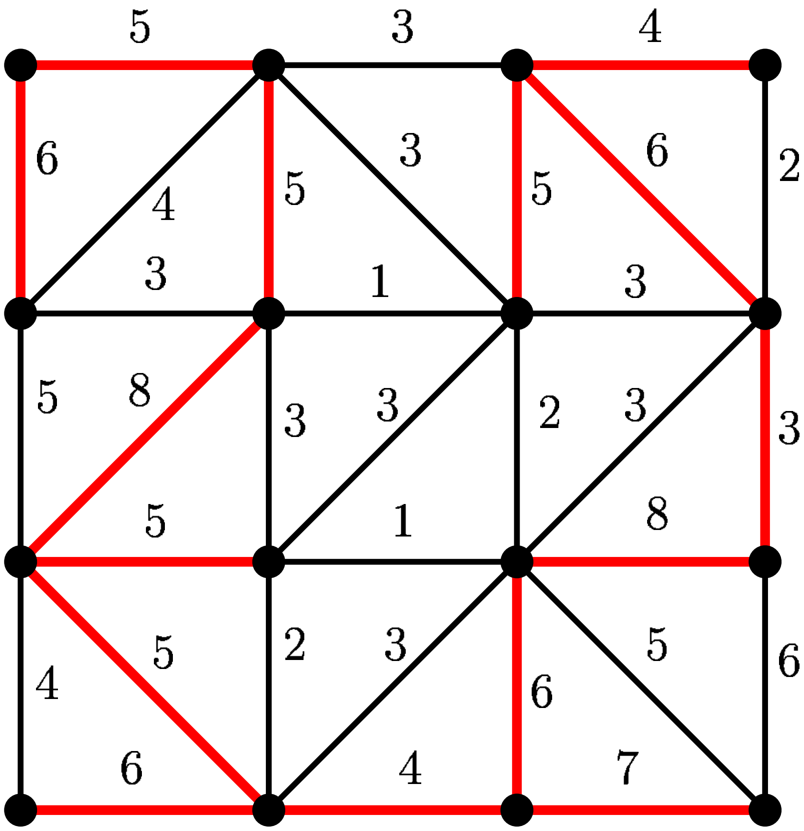
5
3
4
6
4
5
3
5
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7



5
3
4
6
4
5
3
5
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7


5
3
4
6
4
5
3
5
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7


5
3
4
6
4
5
3
5
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
5
2
3
6
5
6
4
6
4
7



最大重みの木 T

貪欲アルゴリズムはなぜ正しいか？

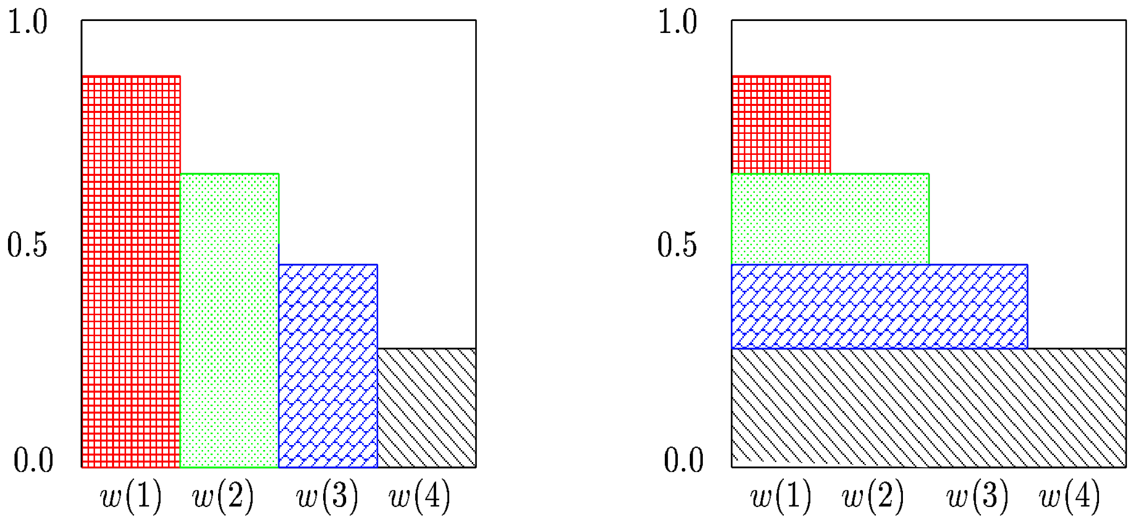
重み $w(e)$ $(e \in E)$ の相異なる値が $k_1 > k_2 > \cdots > k_p$ であるとして,

$$F_i = \{e \in E \mid w(e) \geq k_i\} \qquad (i = 1, 2, \cdots, p)$$

とおくと，重み関数(ベクトル)w は

$$w = (k_1 - k_2)\chi_{F_1} + \cdots + (k_{p-1} - k_p)\chi_{F_{p-1}} + k_p\chi_{F_p}$$

と表現される. ただし，χ_{F_i} は，F_i 上で 1，$E \setminus F_i$ 上で 0 の値を取る関数(特性ベクトル)である.



全域木 T の重みは,

$$\sum_{e\in T} w(e) = (k_1-k_2)\sum_{e\in T}\chi_{F_1}(e) + \cdots + (k_{p-1}-k_p)\sum_{e\in T}\chi_{F_{p-1}}(e) + k_p\sum_{e\in T}\chi_{F_p}(e)$$

$$= (k_1-k_2)|T\cap F_1| + \cdots + (k_{p-1}-k_p)|T\cap F_{p-1}| + k_p|T\cap F_p|$$

(注意: $F_1 \subset F_2 \subset \cdots \subset F_{p-1} \subset F_p (= E)$)

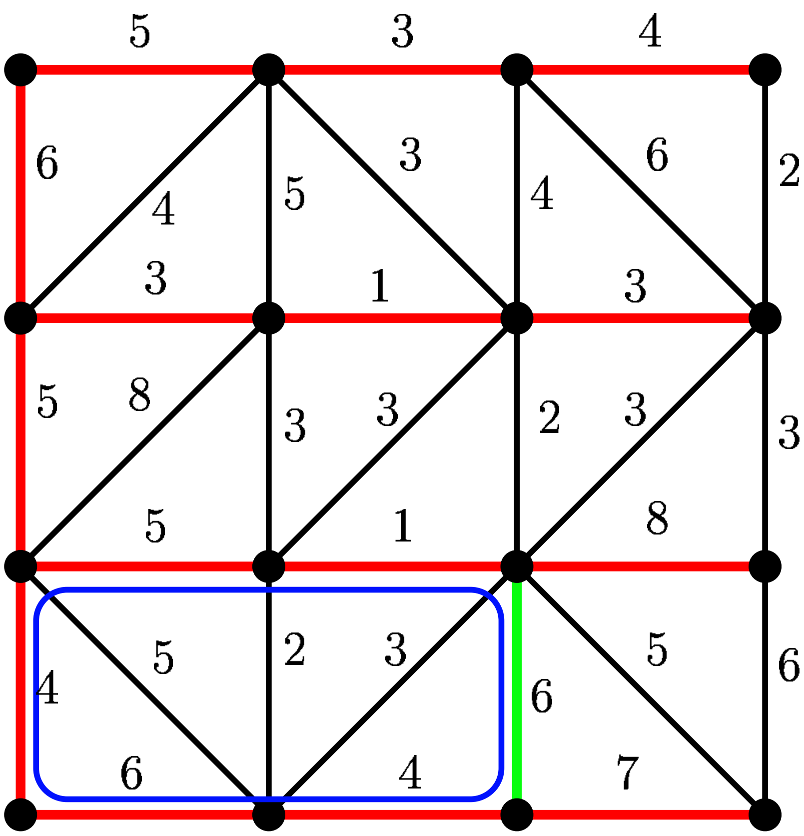
T^σ **は,** $|T\cap F_1|, \cdots, |T\cap F_{p-1}|, |T\cap F_p|$ **を同時に最大化する.**

(⟵ **マトロイド構造**)

ゆえに, T^σ は, 最大木である.

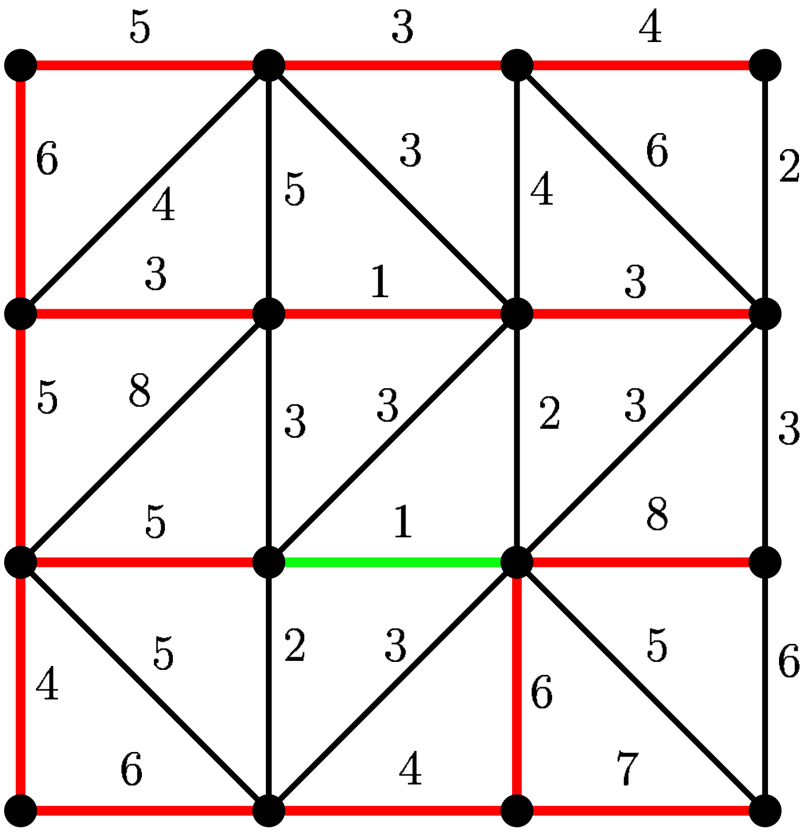
逆に, 任意の最大木 T^* は, $|T\cap F_1|, \cdots, |T\cap F_p|$ を同時に最大化する. よって, ある単調非増加な重みの並べ方に対する貪欲アルゴリズムの解になっている. □

木の初等変換



基本閉路 (基本サイクル)

木の初等変換



最大木の特徴づけ

定理：任意の (大域木)T に対して，T が重み w に関する最大木であるための必要十分条件は、

(*) 木 T のすべての初等変換 $T \cup \{e\} \setminus \{e'\}$ によってその重みが大きくならない，すなわち，$w(e) \le w(e')$ であることである．

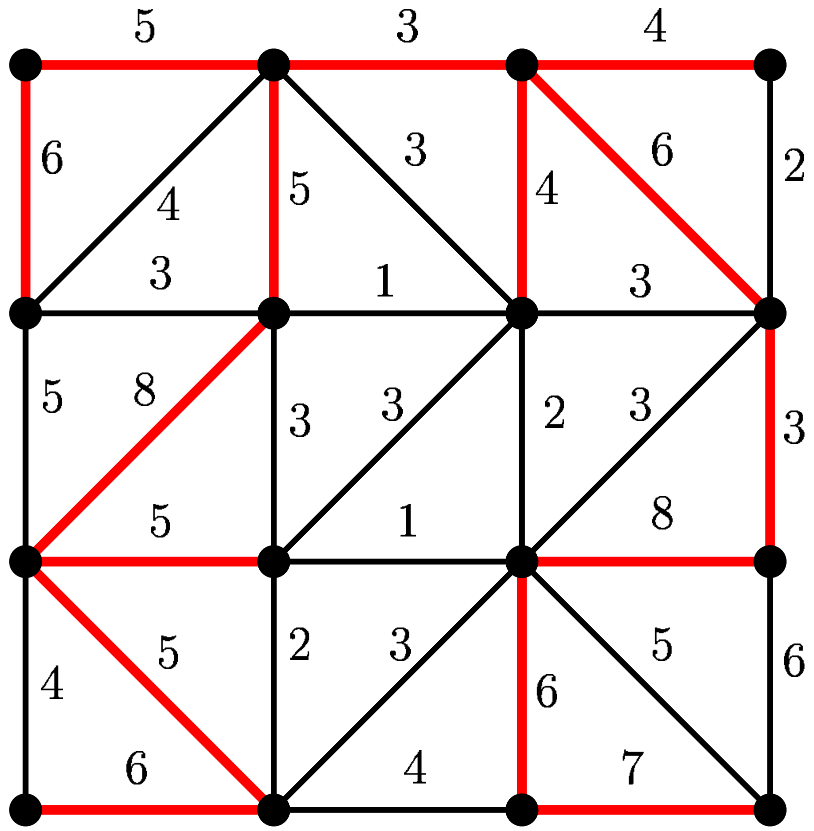
□

(注) (*) が成り立つとき，貪欲アルゴリズムで T が求められるような並び $w(e_1) \ge w(e_2) \ge \cdots \ge w(e_n)$ が存在する．

$\mathrm{O}(|E|^2)$ **個の近傍における局所最適性＝大域最適性　!!**

最大木問題

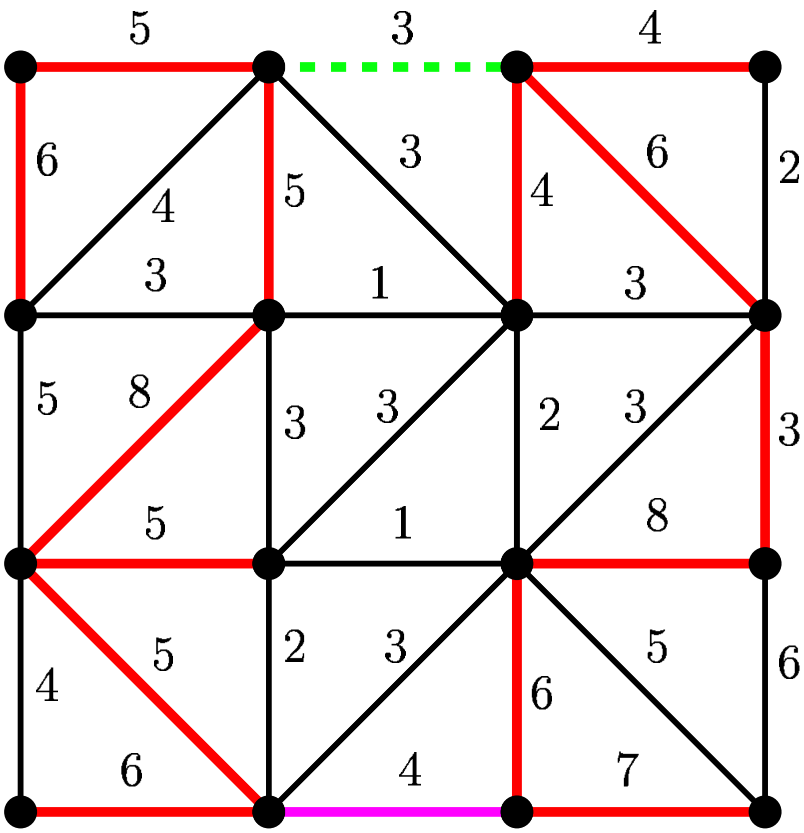
連結な**グラフ** $G = (V, E)$ と**枝重み関数** $w : E \to \mathbf{R}$



重みの総和 $w(T) = 81$　　**最適か？**

最大木問題

連結な**グラフ** $G = (V, E)$ と**枝重み関数** $w : E \to \mathbf{R}$



重みの総和 $w(T) = 82$　　**最適か？**

最大木の特徴づけ

定理：任意の(大域木)Tに対して，Tが重みwに関する最大木であるための必要十分条件は、

(*) 木Tのすべての初等変換$T \cup \{e\} \setminus \{e'\}$によってその重みが大きくならない，すなわち，$w(e) \leq w(e')$であることことである．

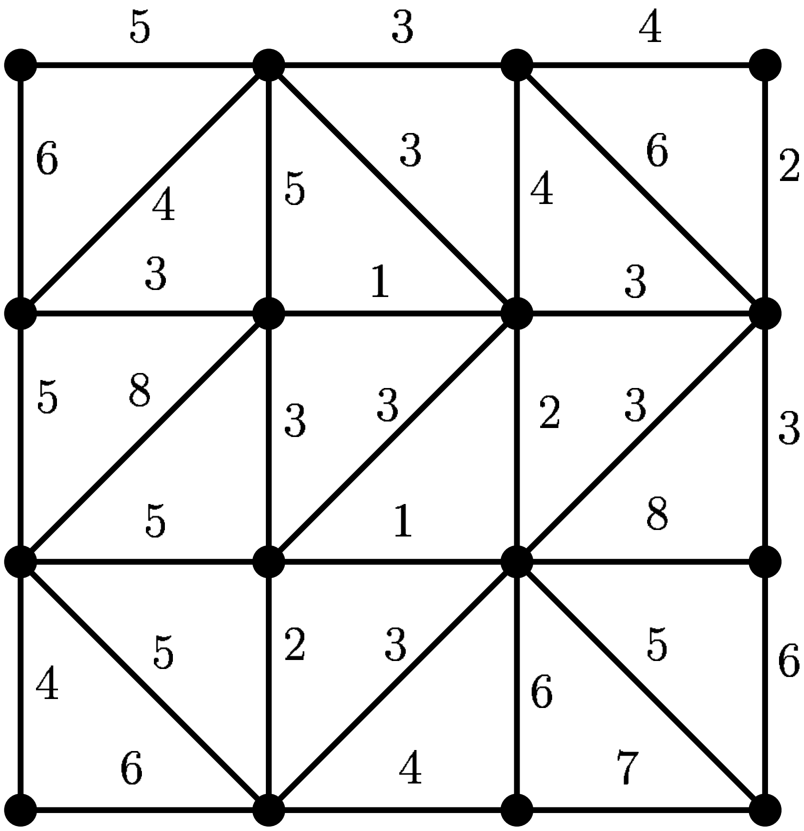
□

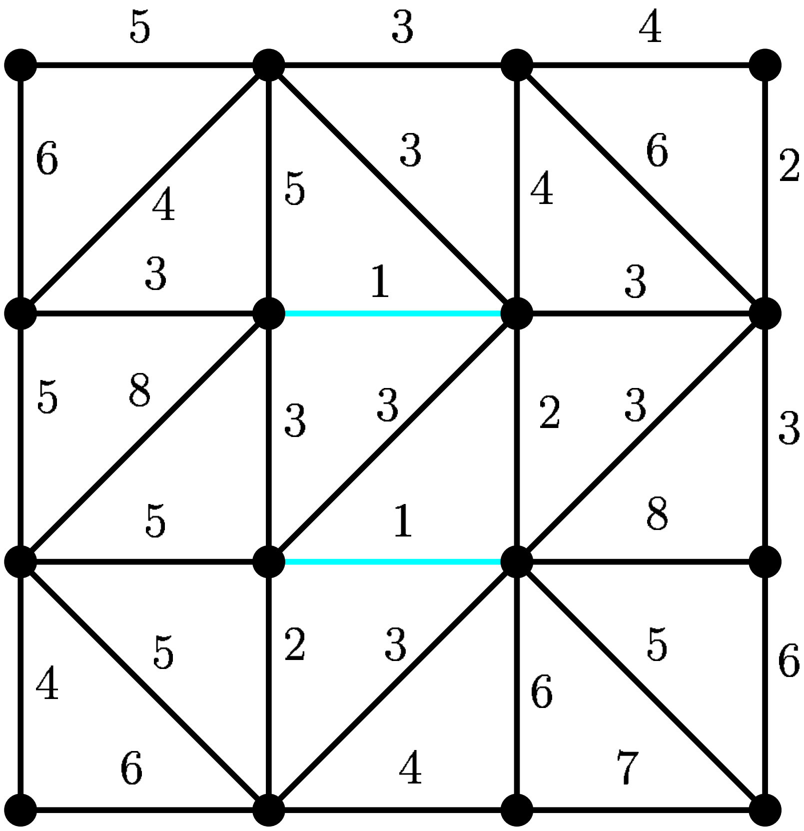
(注) (*) が成り立つとき，貪欲アルゴリズムでTが求められるような並び $w(e_1) \geq w(e_2) \geq \cdots \geq w(e_n)$ が存在する．

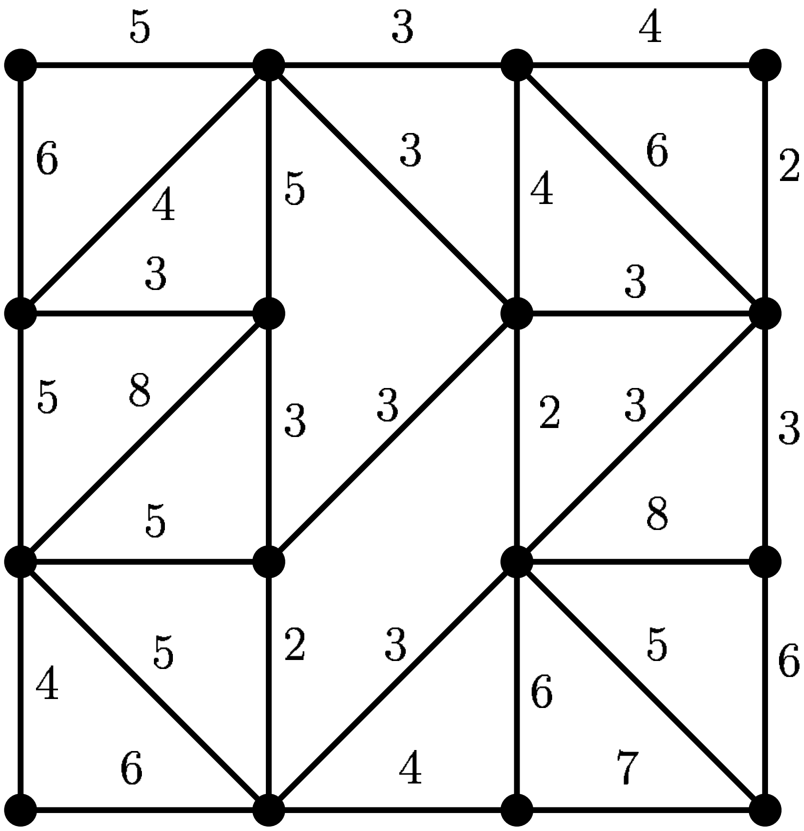
$\mathrm{O}(|E|^2)$ **個の近傍における局所最適性＝大域最適性　!!**

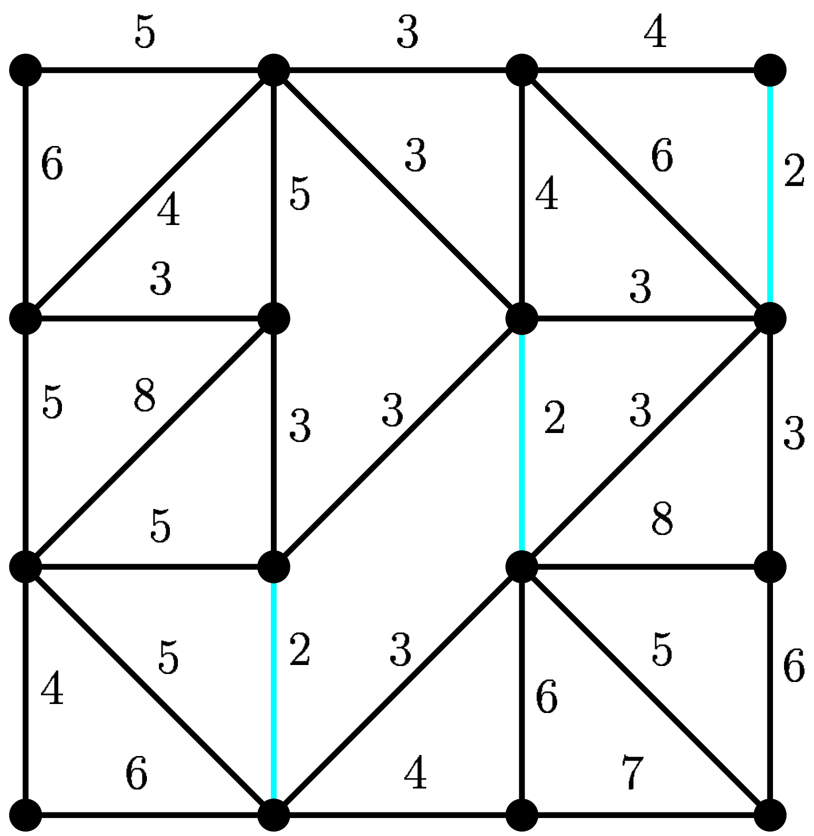
双対貪欲アルゴリズム $\longleftarrow$ マトロイドの双対性

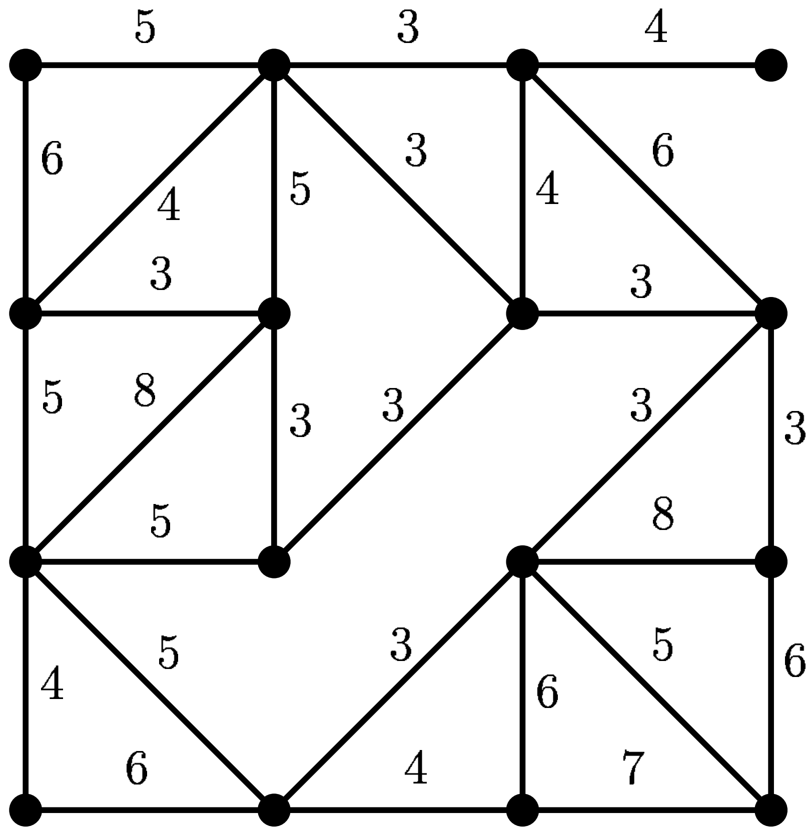
(次ページ以降に例で)

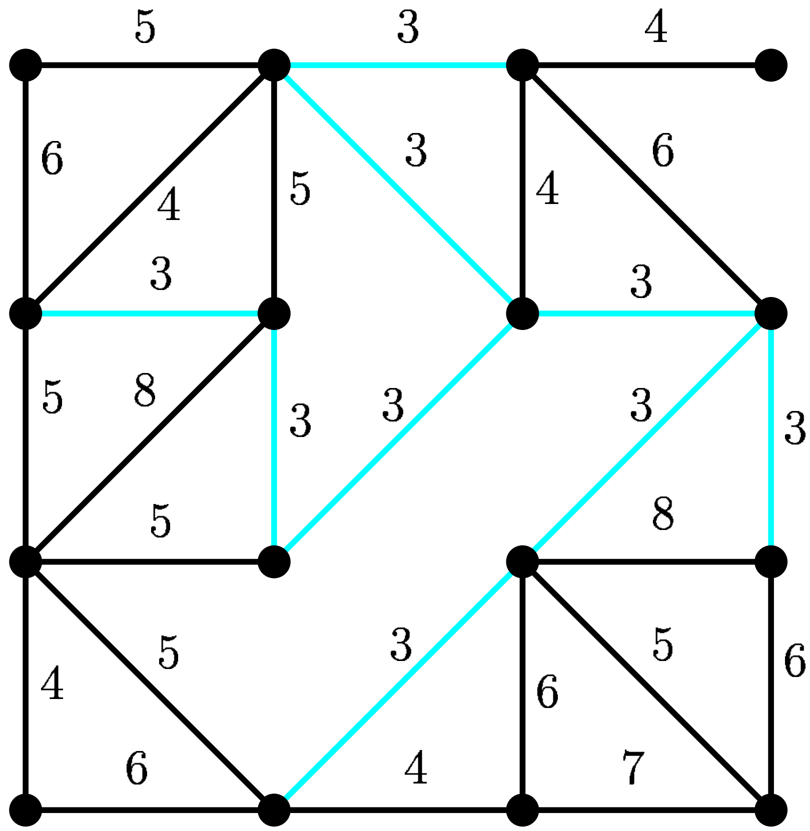

5
3
4
6
4
5
3
4
6
2
3
1
3
5
8
3
3
2
3
3
5
1
8
5
2
3
5
6
4
6
6
4
7

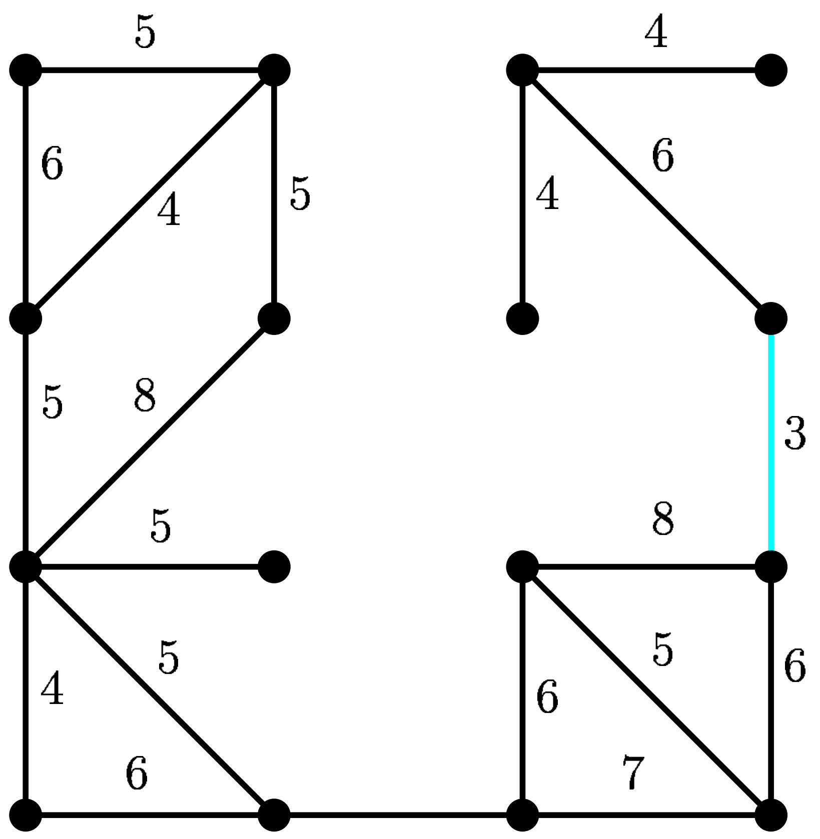

5
3
4
6
3
6
2
5
4
4
3
1
3
5
8
3
3
2
3
3
5
1
8
4
5
2
3
6
5
6
6
4
7

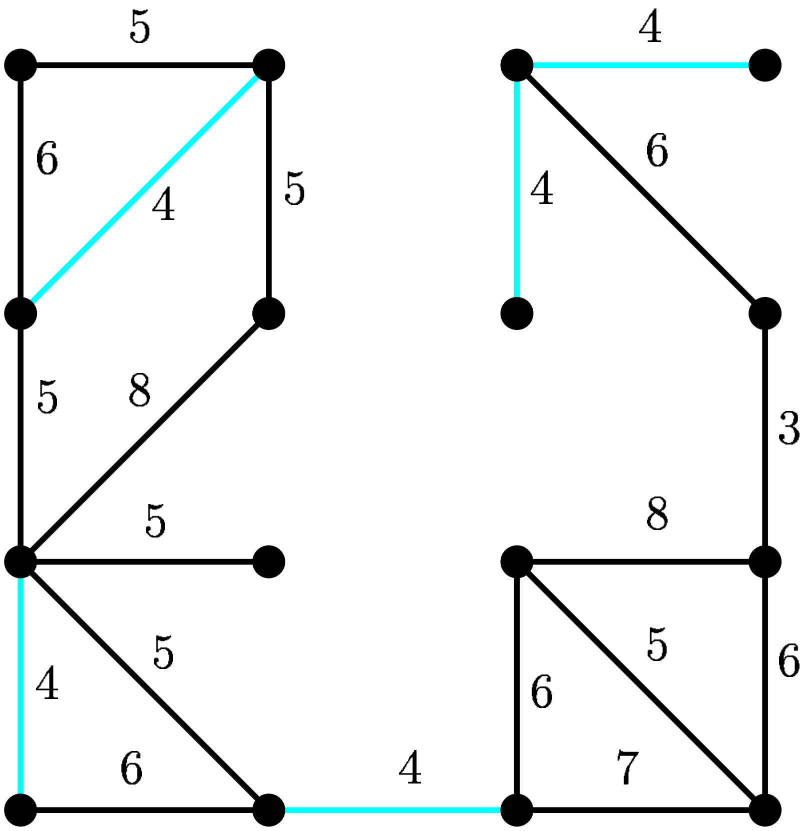

5
3
4
6
3
6
2
4
5
4
3
3
5
8
3
3
2
3
3
5
8
5
2
3
5
6
4
6
6
4
7

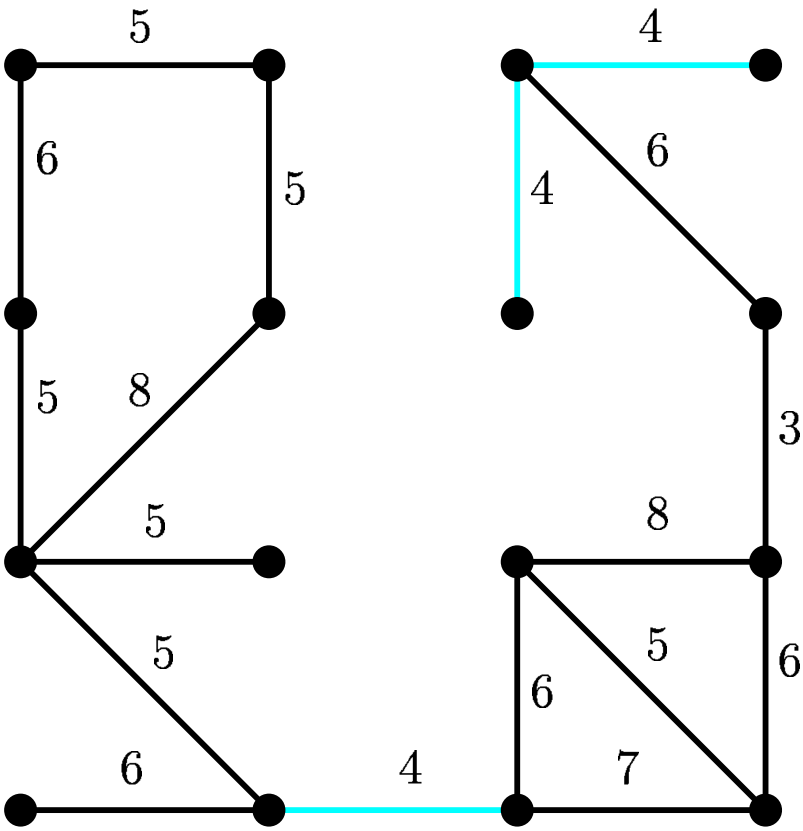

5
3
4
6
4
5
3
4
6
2
3
3
5
8
3
3
2
3
3
5
8
4
5
2
3
5
6
6
6
4
7

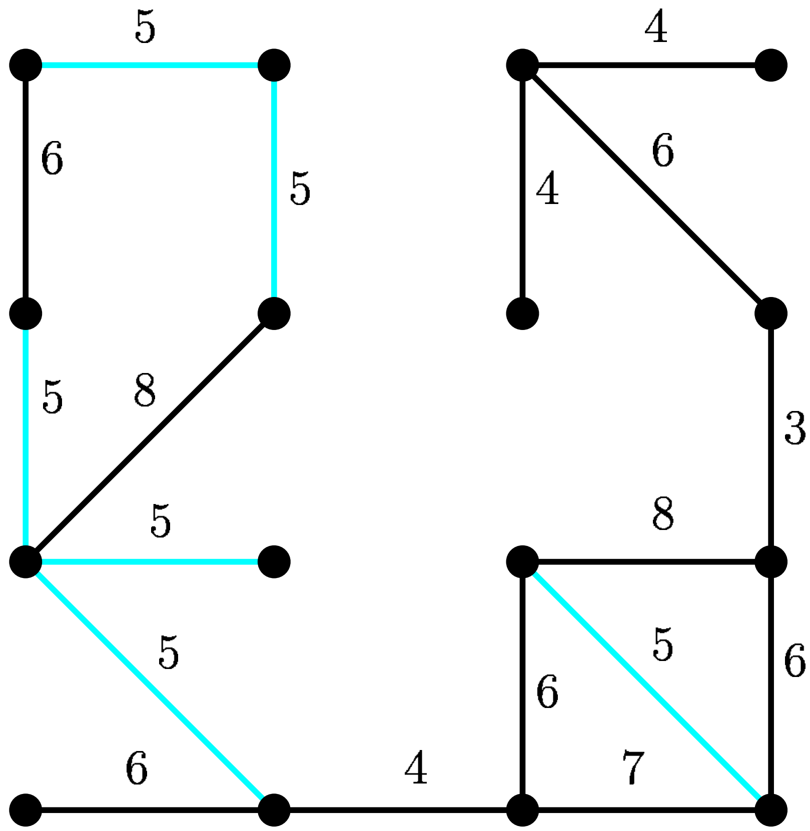

5
3
4
6
4
5
3
4
6
3
3
5
8
3
3
3
3
5
8
5
3
4
5
6
5
6
6
4
7

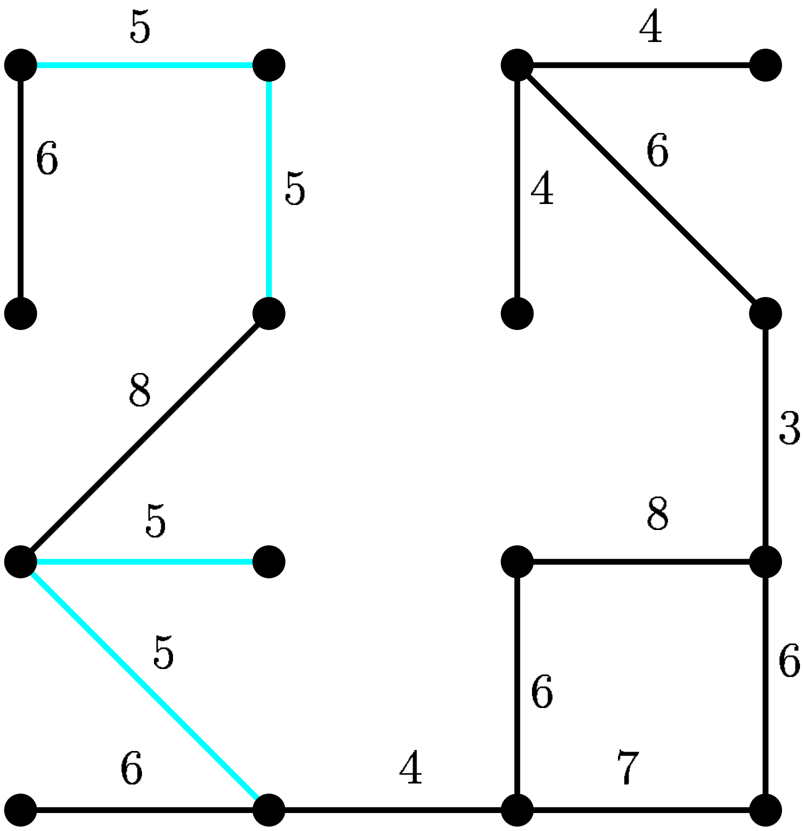

5
3
4
6
4
5
3
4
6
3
3
5
8
3
3
3
3
5
8
4
5
3
5
6
6
6
4
7


5
6
4
5
4
6
4
5
8
3
5
8
5
4
5
6
6
6
7

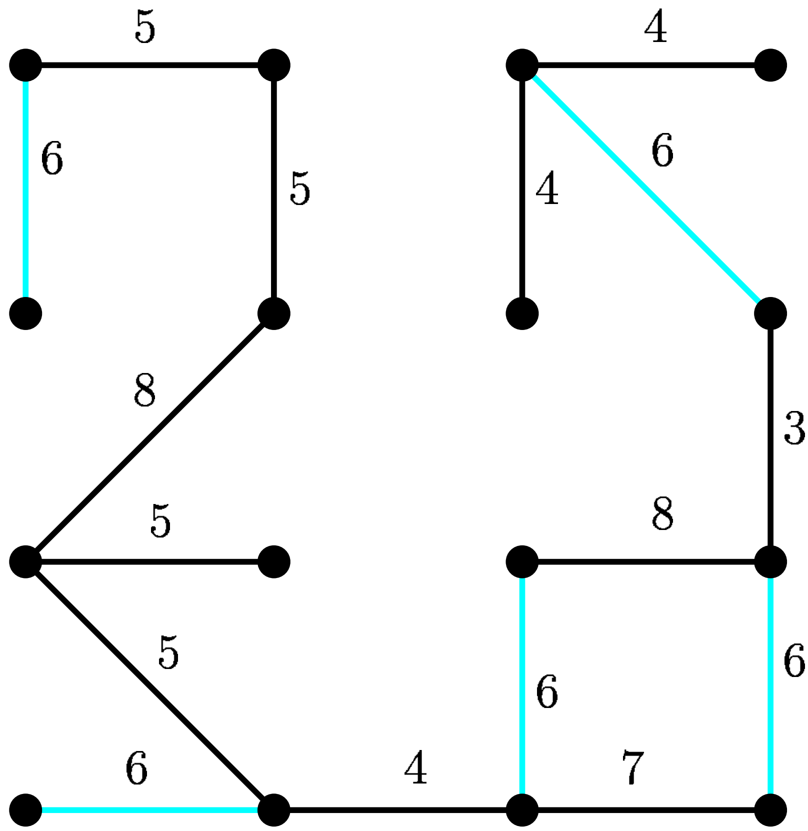

5
4
6
4
5
4
6
5
8
3
5
8
4
5
5
6
6
6
4
7
6

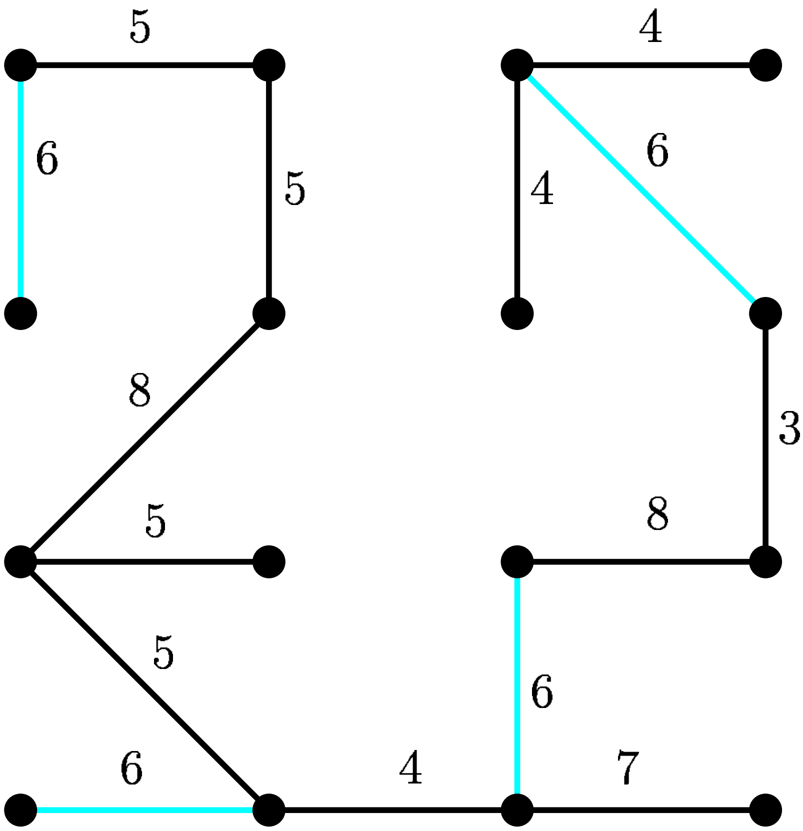

5
6
5
5
8
5
5
6
4
4
4
6
3
8
5
6
6
7


5
4
6
5
6
4
5
8
3
5
8
5
5
6
6
6
4
7



5
4
6
5
4
6
8
3
5
8
5
6
6
6
4
7

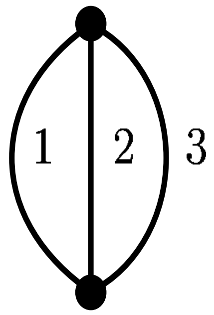

5
4
6
5
4
6
8
3
5
8
5
6
6
6
4
7

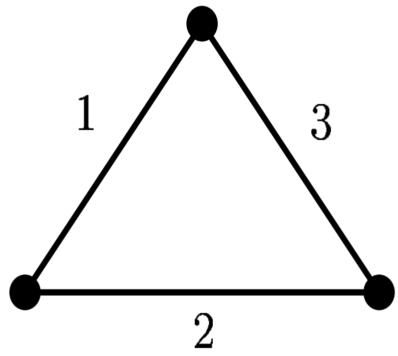


全域木になって，最大重み全域木が見つかった！

最小重みの補木 (削除された枝集合) が見つかった！

木族の多面体表現



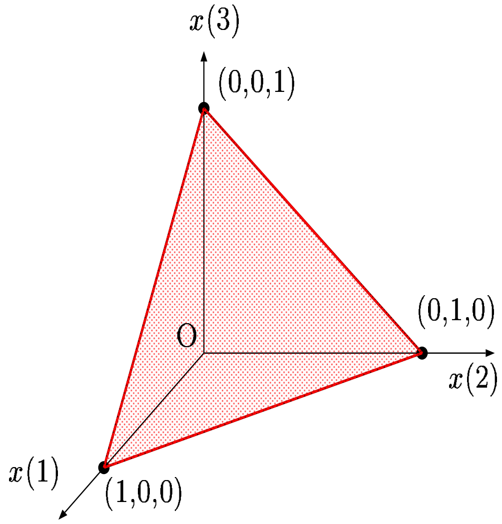


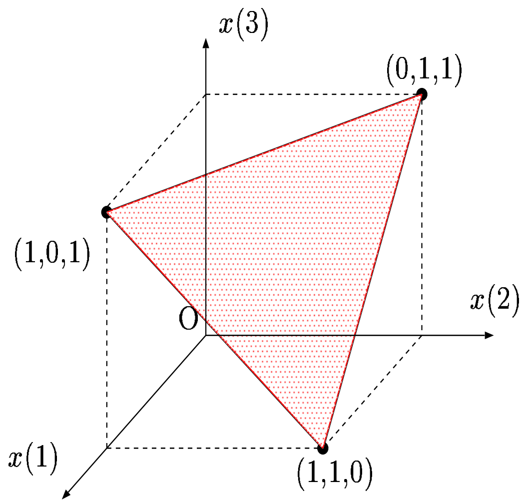
$T:\quad (1,0,0),\ \ (0,1,0),\ \ (0,0,1)$

$T:\quad (1,1,0),\ \ (1,0,1),\ \ (0,1,1)$

(特性ベクトル)

木族の多面体表現





$$\mathrm{B}(f) = \{x \in \mathbf{R}^E \mid \forall X \subset E:\ x(X) \leq f(X),\ x(E) = f(E)\}$$

定理：$\mathrm{B}(f)$ の端点の全体＝(全域) 木の特性ベクトルの全体. □

定理：$\mathrm{B}(f)$ の隣接する端点に対応する (全域) 木 T と T' は，木の初等変換で移り合える二つの木である. □

系：$\mathrm{B}(f)$ の辺ベクトルは，$(0,\cdots,0,\pm1,0,\cdots,0,\mp1,0,\cdots,0)$ の形である. □

(注) $\mathrm{B}(f)$ は**基多面体**と呼ばれる. 辺ベクトルの数は，$\mathrm{O}(|E|^2)$ である.
(**辺多項式な**多面体)

重み関数 $w : E \to [0,1]$

T^w: 重み w に関する最大木

$\hat{f}(w) = w(T^w) \bigl(= \sum_{e \in T^w} w(e)\bigr)$ 　(T^w の重み)

E の順列 $\sigma = (e_1, e_2, \cdots, e_n)$ に対して，

$$\Delta(\sigma): \quad 1 \geq w(e_1) \geq w(e_2) \geq \cdots \geq w(e_n) \geq 0$$

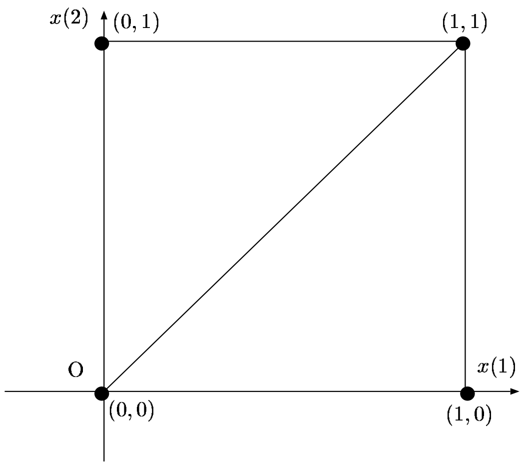
を満たすすべての重み w に対して木 T^σ が存在して，T^σ が重み w に関する最大木となる. ($\longleftarrow$ 貪欲アルゴリズム)

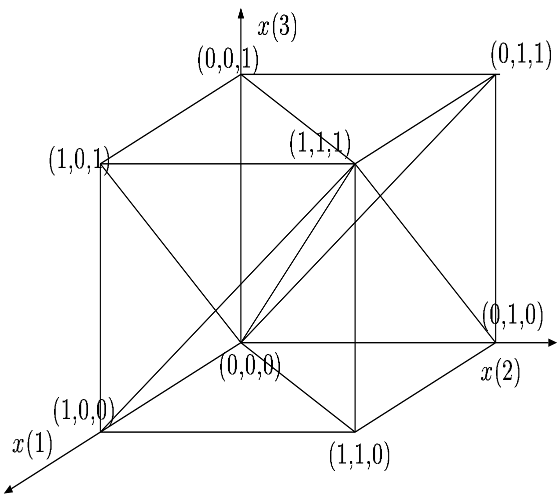
よって，$\hat{f}(w) = \sum_{e \in T^\sigma} w(e)$ であるから，

(*) $\hat{f}$ は各領域 (単体)$\Delta(\sigma)$ 上で線形な関数である.

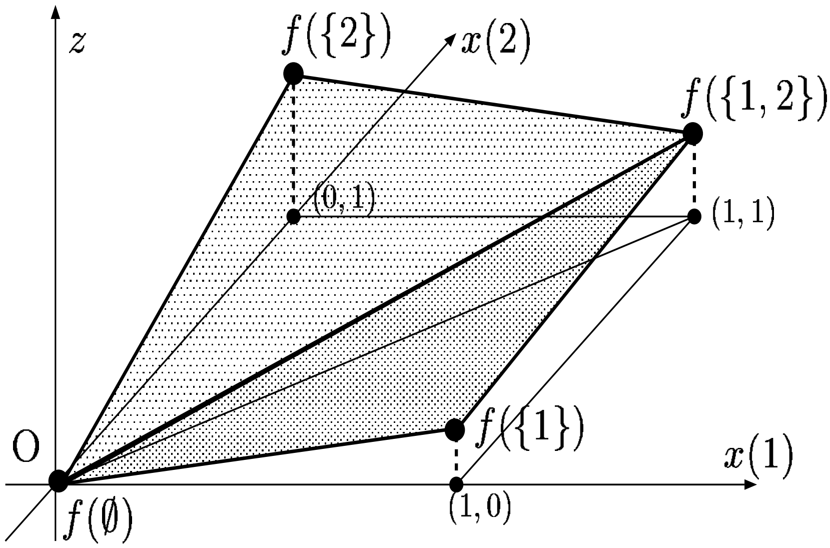
$$\Delta(\sigma): \quad 1 \geq w(e_1) \geq w(e_2) \geq \cdots \geq w(e_n) \geq 0$$

$n!$個の順列σに対応する領域(単体)$\Delta(\sigma)$の集まりは，単位立方体$[0,1]^E$の単体分割を与える．





任意の集合関数 $f : 2^E \to \mathbf{R}$ を，各単体 $\Delta(\sigma)$ 上で線形な関数になるように単位立方体 $[0,1]^E$ 上に拡張した関数 $\hat{f}$ は，f の **Lovász 拡張**と呼ばれる．



定理：任意の集合関数 $f : 2^E \to \mathbf{R}$ が劣モジュラ関数であるための必要十分条件は，その Lovász 拡張 $\hat{f}$ が凸関数であることである． □

隣接する二つの単体に対応する順列は

$$\sigma_1 = (i_1, \cdots, i_{k-1}, i_k, i_{k+1}, i_{k+2}, \cdots, i_n),$$

$$\sigma_2 = (i_1, \cdots, i_{k-1}, i_{k+1}, i_k, i_{k+2}, \cdots, i_n)$$

の形で与えられる (共有する面の方程式は，$x(i_k) - x(i_{k+1}) = 0$).

$$S_\ell = \{i_p \mid p = 1, \cdots, \ell\} \qquad (\ell = 1, \cdots, n)$$

とおく. σ_1, σ_2 に対応して**貪欲アルゴリズム**で求められる (全域) 木を $T^{\sigma_1}, T^{\sigma_2}$ とすると，**マトロイド構造**によって,

$$S_\ell \cap T^{\sigma_1} = S_\ell \cap T^{\sigma_2} \qquad (\ell = 1, \cdots, k-1, k+1, \cdots, n)$$

$$|\{i_k, i_{k+1}\} \cap T^{\sigma_1}| = |\{i_k, i_{k+1}\} \cap T^{\sigma_2}|$$

が成り立つ. よって，$T^{\sigma_1} \neq T^{\sigma_2}$ であるならば,

$$i_k \in T^{\sigma_1}, \quad i_{k+1} \notin T^{\sigma_1}$$

$$i_k \notin T^{\sigma_2}, \quad i_{k+1} \in T^{\sigma_2}$$

(注) $b_1 = \chi_{T^{\sigma_1}}, b_2 = \chi_{T^{\sigma_2}}$ は，基多面体 $\mathrm{B}(f)$ の端点であり，$b_1 \neq b_2$ のとき，これらの二つの端点は隣接する. □

劣モジュラ性・劣モジュラ関数 $f : 2^E \to \mathbf{R}$

$\Updownarrow$

基多面体 $\mathrm{B}(f)$ (辺ベクトルが $(0, \cdots, 0, \pm 1, 0, \cdots, 0, \mp 1, 0, \cdots, 0)$)

$\Updownarrow$

貪欲アルゴリズム (重みの大小関係だけで最適解が決まる)

$\Updownarrow$

Lovász 拡張 $\hat{f}$ が凸 (各単体 $\Delta(\sigma)$ 上で線形な凸関数)

劣モジュラ性・劣モジュラ関数 $f : 2^E \to \mathbf{R}$

$\Updownarrow$

基多面体 $\mathrm{B}(f)$ (辺ベクトルが $(0, \cdots, 0, \pm 1, 0, \cdots, 0, \mp 1, 0, \cdots, 0)$)

$\Updownarrow$

貪欲アルゴリズム (重みの大小関係だけで最適解が決まる)

$\Updownarrow$

Lovász 拡張 $\hat{f}$ が凸 (各単体 $\Delta(\sigma)$ 上で線形な凸関数)

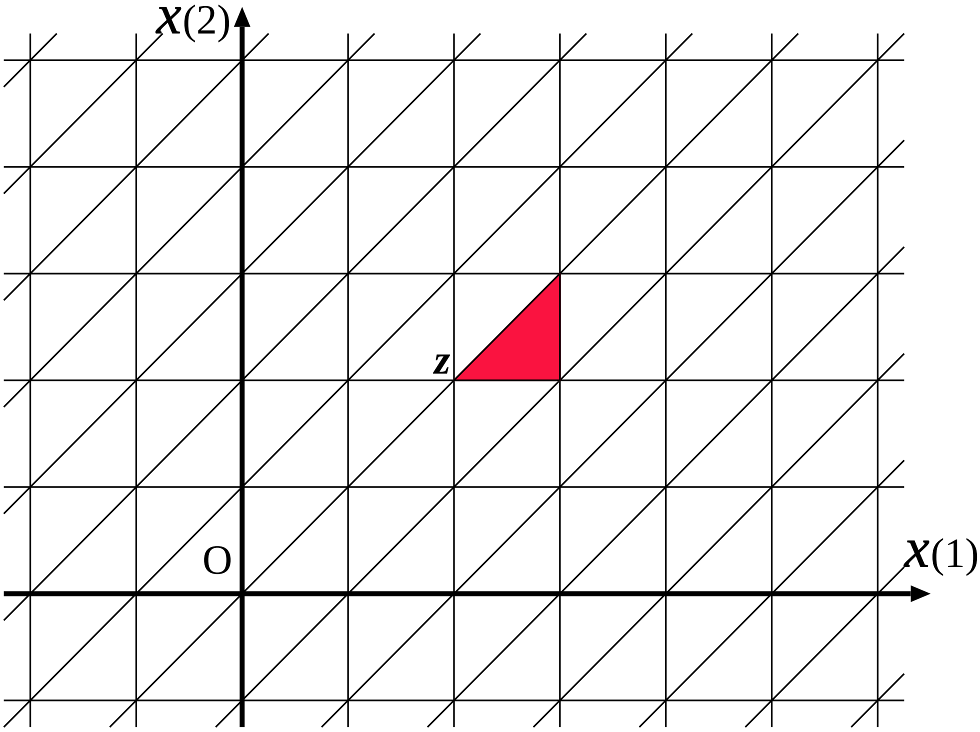
整数格子点上の関数に対して，以上の議論を拡張可能

$\Downarrow$

室田一雄氏 (東大) による「離散凸解析」 (L$^\natural$ 凸関数)

Coxeter-Freudenthal 単体分割上の凸関数

$\longrightarrow$ L$^\natural$ 凸関数　(Murota (1996), Favati-Tardella (1990))



$$f : \mathbf{Z}^n \to \mathbf{R}$$

$$f(x) + f(y) \geq f(x \vee y) + f(x \wedge y) \qquad (x,\ y \in \mathbf{Z}^n)$$

(この $\mathbf{Z}^n$ 上の劣モジュラ性だけからは f の大域的な凸性は出てこない。)

劣モジュラ関数 $f : 2^E \to \mathbf{R}$

$\Updownarrow$

Lovász 拡張 (凸関数) $\hat{f}$

優モジュラ関数 $g : 2^E \to \mathbf{R}$

$\Updownarrow$

Lovász 拡張 (凹関数) $\hat{g}$

通常の凸関数と凹関数の分離定理＋整数性

$\Downarrow$

Lovász 拡張 (凸関数)$\hat{f}$ と Lovász 拡張 (凹関数)$\hat{g}$ の分離定理

$\Updownarrow$

基多面体の交わり定理

(最大・最小定理，整数性，効率的アルゴリズムをもつ)

$\Downarrow$

劣モジュラ・フロー問題

(効率良く解ける離散最適化・組合せ最適化の広いクラスをモデル化する．)

通常の凸関数と凹関数の分離定理＋整数性

$\Downarrow$

Lovász 拡張 (凸関数)$\hat{f}$ と Lovász 拡張 (凹関数)$\hat{g}$ の分離定理

$\Updownarrow$

基多面体の交わり定理

(最大・最小定理，整数性，効率的アルゴリズムをもつ)

$\Downarrow$

劣モジュラ・フロー問題

(効率良く解ける離散最適化・組合せ最適化の広いクラスをモデル化する．)

劣モジュラ構造：　多くの物や人が関わる複雑なシステムの最適化にしばしば顔を出す.

$\Longrightarrow$ **効率的なアルゴリズムの構成**

参考図書など

1. 伊理・藤重・大山: 「グラフ・ネットワーク・マトロイド」(講座・数理計画第 7 巻), 産業図書 (1986).
2. 藤重 悟：「グラフ・ネットワーク・組合せ論」(工系数学講座 18 巻)，共立出版 (2002)
3. 藤重 悟：「劣モジュラ構造と離散凸性」, 数学入門公開講座 (数理解析研究所，2005) (テキストは数理解析研ホームページで公開).
4. S. Fujishige: *Submodular Functions and Optimization* Second Edition (Annals of Discrete Mathematics, Vol. 58, Elsevier, 2005).
5. 室田 一雄：「離散凸解析」, 共立出版, 2001.
6. K. Murota: *Discrete Convex Analysis* (SIAM Monographs on Discrete Mathematics and Applications **10**, SIAM, 2003).
7. 田村 明久：「離散凸解析とゲーム理論」(オペレーションズ・リサーチ 3)，朝倉書店 (2009).

e-mail: fujishig@kurims.kyoto-u.ac.jp

URL: http://www.kurims.kyoto-u.ac.jp/~fujishig/