離散最適化基礎論 第 1 回
組合せ最適化におけるマトロイドの役割

岡本 吉央
okamotoy@uec.ac.jp

電気通信大学

2015 年 10 月 9 日

最終更新：2015 年 10 月 9 日　21:45

概要

主題

離散最適化のトピックの 1 つとして
組合せ最適化におけるマトロイドの役割を取り上げ，

- マトロイドとは何か？
- マトロイドがなぜ役に立つのか？
- マトロイドがどう役に立つのか？

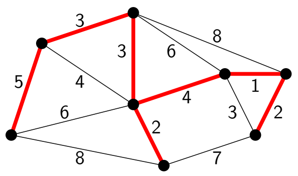
について，数理的側面と計算的側面の双方を意識して講義する

なぜ講義で取り扱う？

- 「組合せ最適化の神髄」だから

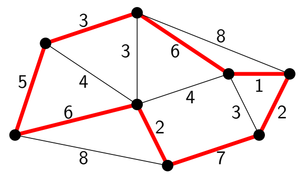
テーマ：組合せ最適化問題の解きやすさ

最小全域木問題：すべての頂点間に経路が存在するような
重み和最小のネットワークを作る



テーマ：組合せ最適化問題の解きやすさ

巡回セールスマン問題：すべての頂点をちょうど一度ずつ通るような
重み和最小の巡回路を作る



テーマ：組合せ最適化問題の解きやすさ

解きやすい問題

- 最小全域木問題
- 最大マッチング問題
- 最小カット問題
- ...

解きにくい問題

- 巡回セールスマン問題
- 最小頂点被覆問題
- 最小彩色問題
- ...

「解きやすい」とは

多項式時間解法が存在する
(参考：アルゴリズム論第一・第二)

「解きにくい」とは

NP 困難性が証明されている
(参考：計算理論)

疑問

どうしてそのような違いが生まれるのか？

⇝ 解きやすい問題が持つ「共通の性質」は何か？

テーマ：解きやすい組合せ最適化問題が持つ「共通の性質」

疑問

どうしてそのような違いが生まれるのか？

⇝ 解きやすい問題が持つ「共通の性質」は何か？

回答

よく分かっていない

しかし，部分的な回答はある

部分的な回答

問題が「マトロイド的構造」を持つと解きやすい

ポイント

効率的アルゴリズムが設計できる背景に「美しい数理構造」がある

この講義では，その一端に触れたい

スケジュール 前半 (予定)

⋆	休講 (卒研準備発表会)	(10/2)
1	組合せ最適化問題におけるマトロイドの役割	(10/9)
⋆	休講 (海外出張)	(10/16)
2	マトロイドの定義と例	(10/23)
3	マトロイドの基と階数関数	(10/30)
4	グラフの全域木	(11/6)
5	マトロイドとグラフの全域木	(11/13)
⋆	休講 (調布祭)	(11/20)
6	マトロイドに対する貪欲アルゴリズム	(11/27)
7	マトロイドのサーキット	(12/4)

注意：予定の変更もありうる

スケジュール 後半 (予定)

⋆	休講 (国内出張)	(12/11)
8	マトロイドに対する操作	(12/18)
9	マトロイドの交わり	(12/25)
⋆	冬季休業	(1/1)
10	マトロイド交わり定理	(1/8)
⋆	休講 (センター試験準備)	(1/15)
11	マトロイド交わり定理：アルゴリズム	(1/22)
12	最近のトピック	(1/29)
⋆	授業等調整日 (予備日)	(2/5)
⋆	期末試験	(2/12?)

注意：予定の変更もありうる

情報

教員
- 岡本 吉央 (おかもと よしお)
- 居室：西 4 号館 2 階 206 号室
- E-mail：okamotoy@uec.ac.jp
- Web：http://dopal.cs.uec.ac.jp/okamotoy/

講義資料
- Web：http://dopal.cs.uec.ac.jp/okamotoy/lect/2015/matroid/
- 注意：資料の印刷等は各学生が自ら行う
- 講義当日の昼 12 時までに，ここに置かれる
- Twitter (@okamoto7yoshio)：置かれたことを知らせる tweet

講義資料

http://dopal.cs.uec.ac.jp/okamotoy/lect/2015/matroid/
- スライド
- 印刷用スライド：8 枚のスライドを 1 ページに収めたもの
- 演習問題

「印刷用スライド」と「演習問題」は各自印刷して持参すると便利

授業の進め方

講義 (80 分)
- スライドと板書で進める
- スライドのコピーに重要事項のメモを取る

演習 (10 分)
- 演習問題に取り組む
- 不明な点は教員に質問する

退室 (0 分) ←重要
- コメント (授業の感想，質問など) を紙に書いて提出する (匿名可)
- コメントとそれに対する回答は (匿名で) 講義ページに掲載される

オフィスアワー：金曜 5 限 (岡本居室か CED)
- 質問など
- ただし，いないときもあるので注意 (注意：情報数理工学セミナー)

演習問題

演習問題の進め方
- 授業のおわり 15 分は演習問題を解く時間
- 残った演習問題は復習・試験対策用
- 注意：「模範解答」のようなものは存在しない

演習問題の種類
- 復習問題：講義で取り上げた内容を反復
- 補足問題：講義で省略した内容を補足
- 追加問題：講義の内容に追加
- 発展問題：少し難しい (かもしれない)

答案の提出
- 演習問題の答案をレポートとして提出してもよい
- レポートには提出締切がある (各回にて指定)
- レポートは採点されない (成績に勘案されない)
- レポートにはコメントがつけられて，返却される
  - 返却された内容については，再提出ができる (再提出締切は原則なし)

評価

期末試験のみによる
- 出題形式
  - 演習問題と同じ形式の問題を 6 題出題する
  - その中の 3 題以上は演習問題として提示されたものと同一である (ただし,「発展」として提示された演習問題は出題されない)
  - 全問に解答する
- 配点：1 題 20 点満点，計 120 点満点
- 成績において，100 点以上は 100 点で打ち切り
- 時間：90 分 (おそらく)
- 持ち込み：A4 用紙 1 枚分 (裏表自筆書き込み) のみ可

教科書・参考書

教科書
- 指定しない

全般的な参考書
- B. コルテ，J. フィーゲン (著)，浅野孝夫，浅野泰仁，小野孝男，平田富夫 (訳),『組合せ最適化 第 2 版』，丸善出版，2012 年.
- W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver, *Combinatorial Optimization*, Wiley, 1997.
- A. Schrijver, *Combinatorial Optimization*, Springer, 2002.
- J. Oxley, *Matroid Theory*, Oxford, 1992.
- その他，研究論文

この講義の約束

- 私語はしない (ただし，演習時間の相談は積極的に OK)
- 携帯電話はマナーモードにする
- この講義と関係のないことを (主に電子機器で) しない
- 音を立てて睡眠しない

約束が守られない場合は退席してもらう場合あり

目次

例 1：ナップサック問題 (knapsack problem)

ナップサック問題　　(森，松井 '04 より)

- いま手元に 4 つの商品が 1 つずつあるとしよう．
- これをナップサックに詰めて街に売りに行くとする．
- それぞれの重さと，売った際の収益は表の通りとする．

商品	1	2	3	4
収入 [万円]	3	4	1	2
重さ [kg]	2	3	1	3

- 商品はどれも，街にもっていけば必ず売れるとする．
- ただし，街にもっていく際ナップサックに積めて運ぶのだが，ナップサックに積載重量制限があり，最大でも 4 kg までしか積めないとする．
- このとき，ナップサックの重量制限以内で総収益を最大にするには，どの荷物を積めていったらよいか？

例 1：ナップサック問題 — 積めるか積めないか (1)

商品の集合 $\{1,2,3,4\}$，ナップサックの積載重量制限は 4 kg

商品	1	2	3	4
収入 [万円]	3	4	1	2
重さ [kg]	2	3	1	3

$\{1,2,3,4\}$ は積めるか　→ 重さの和 $= 2+3+1+3 = 9$　→ 積めない
$\{1,2,3\}$ は積めるか　→ 重さの和 $= 2+3+1 = 6$　→ 積めない
$\{1,2,4\}$ は積めるか　→ 重さの和 $= 2+3+3 = 8$　→ 積めない
$\{1,3,4\}$ は積めるか　→ 重さの和 $= 2+1+3 = 6$　→ 積めない
$\{2,3,4\}$ は積めるか　→ 重さの和 $= 3+1+3 = 7$　→ 積めない
$\{1,2\}$ は積めるか　→ 重さの和 $= 2+3 = 5$　→ 積めない
$\{1,3\}$ は積めるか　→ 重さの和 $= 2+1 = 3$　→ 積める
$\{1,4\}$ は積めるか　→ 重さの和 $= 2+3 = 5$　→ 積めない

例 1：ナップサック問題 — 積めるか積めないか (2)

商品の集合 $\{1,2,3,4\}$，ナップサックの積載重量制限は 4 kg

商品	1	2	3	4
収入 [万円]	3	4	1	2
重さ [kg]	2	3	1	3

$\{2,3\}$ は積めるか　→ 重さの和 $= 3+1 = 4$　→ 積める
$\{2,4\}$ は積めるか　→ 重さの和 $= 3+3 = 6$　→ 積めない
$\{3,4\}$ は積めるか　→ 重さの和 $= 1+3 = 4$　→ 積める
$\{1\}$ は積めるか　→ 重さの和 $= 2$　→ 積める
$\{2\}$ は積めるか　→ 重さの和 $= 3$　→ 積める
$\{3\}$ は積めるか　→ 重さの和 $= 1$　→ 積める
$\{4\}$ は積めるか　→ 重さの和 $= 3$　→ 積める
$\emptyset$ は積めるか　→ 重さの和 $= 0$　→ 積める

例 1：ナップサック問題 — 許容集合

商品の集合 $\{1,2,3,4\}$，ナップサックの積載重量制限は 4 kg

商品	1	2	3	4
収入 [万円]	3	4	1	2
重さ [kg]	2	3	1	3

- この問題の許容集合 (feasible set) は

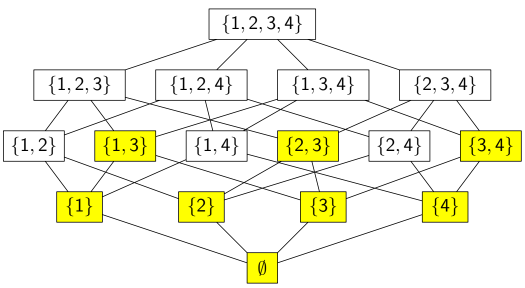
$$\mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1,3\}, \{2,3\}, \{3,4\}\}$$

- $\mathcal{F}$ の中で収入和が最も大きいものを選びたい (目的は最大化)

(許容集合の定義は後述)

例 1：ナップサック問題 — 許容集合 (図示)

$\mathcal{F}$ の要素を黄色で表している

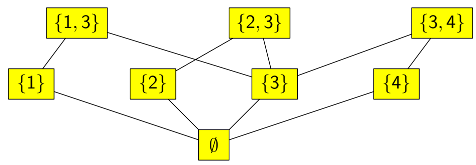


$$\mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1,3\}, \{2,3\}, \{3,4\}\}$$

ハッセ図 (Hasse diagram)

例 1：ナップサック問題 — 許容集合 (図示)

$\mathcal{F}$ の要素を黄色で表している



$$\mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1,3\}, \{2,3\}, \{3,4\}\}$$

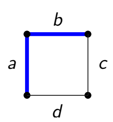
ハッセ図 (Hasse diagram)

例 2：最大重みマッチング問題 — マッチングの定義

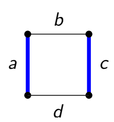
無向グラフ $G = (V, E)$　　(V：頂点集合，E：辺集合)

マッチングとは？

G のマッチング (matching) とは，G の辺部分集合 $M \subseteq E$ で，任意の頂点 $v \in V$ に対して，v に接続する M の辺が 1 つ以下であるもの



$\{a, b\}$ はマッチングではない



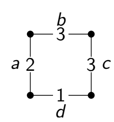
$\{a, c\}$ はマッチングである

例 2：最大重みマッチング問題

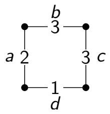
無向グラフ $G = (V, E)$　　(V：頂点集合，E：辺集合)

最大重みマッチング

無向グラフ $G = (V, E)$ と各辺 $e \in E$ の重み $w(e)$ が与えられたとき，G のマッチングの中で，重み和が最大のものを見つける問題

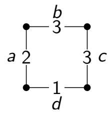


例 2：最大重みマッチング問題 — マッチングであるかないか (1)



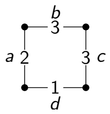
$\{a, b, c, d\}$ はマッチングか → マッチングではない
$\{a, b, c\}$ はマッチングか → マッチングではない
$\{a, b, d\}$ はマッチングか → マッチングではない
$\{a, c, d\}$ はマッチングか → マッチングではない
$\{b, c, d\}$ はマッチングか → マッチングではない
$\{a, b\}$ はマッチングか → マッチングではない
$\{a, c\}$ はマッチングか → マッチングである
$\{a, d\}$ はマッチングか → マッチングではない

例 2：最大重みマッチング問題 — マッチングであるかないか (2)



$\{b, c\}$ はマッチングか → マッチングではない
$\{b, d\}$ はマッチングか → マッチングである
$\{c, d\}$ はマッチングか → マッチングではない
$\{a\}$ はマッチングか → マッチングである
$\{b\}$ はマッチングか → マッチングである
$\{c\}$ はマッチングか → マッチングである
$\{d\}$ はマッチングか → マッチングである
$\emptyset$ はマッチングか → マッチングである

例 2：最大重みマッチング問題 — 許容集合



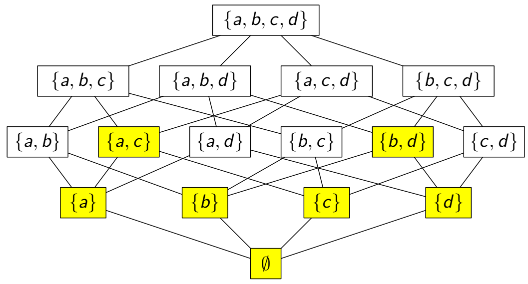
- この問題の許容集合は

$$\mathcal{F} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{b, d\}\}$$

- $\mathcal{F}$ の中で重み和が最も大きいものを選びたい (目的は最大化)

例 2：最大重みマッチング問題 — 許容集合 (図示)

$\mathcal{F}$ の要素を黄色で表している

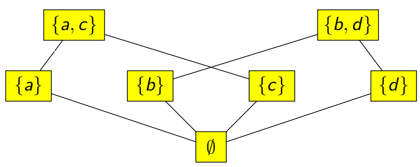


$$\mathcal{F} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{b, d\}\}$$

ハッセ図 (Hasse diagram)

例 2：最大重みマッチング問題 — 許容集合 (図示)

$\mathcal{F}$ の要素を黄色で表している



$$\mathcal{F} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{b, d\}\}$$

ハッセ図 (Hasse diagram)

組合せ最適化問題：定義

組合せ最適化問題：設定

- 非空な有限集合 E
- 有限集合族 $\mathcal{F} \subseteq 2^E$
- 重み関数 $w \colon E \to \mathbb{R}_+$

2^E は E の冪集合，$\mathbb{R}_+$ は非負実数全体の集合

組合せ最適化問題：定義

次のような X を見つける問題

$$\begin{array}{ll} \text{maximize} & \sum_{e \in X} w(e) \\ \text{subject to} & X \in \mathcal{F} \end{array}$$

組合せ最適化問題：用語 (1)

組合せ最適化問題：設定

- 非空な有限集合 E 台集合 (ground set)
- 有限集合族 $\mathcal{F} \subseteq 2^E$ 許容集合 (feasible set)
- 重み関数 $w \colon E \to \mathbb{R}_+$

2^E は E の冪集合，$\mathbb{R}_+$ は非負実数全体の集合

組合せ最適化問題：定義

次のような X を見つける問題

$$\begin{array}{ll} \text{maximize} & \sum_{e \in X} w(e) \\ \text{subject to} & X \in \mathcal{F} \end{array}$$

組合せ最適化問題：用語 (2)

組合せ最適化問題：定義

次のような X を見つける問題

$$\begin{array}{ll} \text{maximize} & \sum_{e \in X} w(e) \\ \text{subject to} & X \in \mathcal{F} \end{array}$$

- $\sum_{e \in X} w(e)$ 目的関数 (objective function)
- $X \in \mathcal{F}$ という条件 制約 (constraint)
- $X \in \mathcal{F}$ という条件を満たす X 許容解 (feasible solution)

組合せ最適化問題：用語 (最適解)

組合せ最適化問題：定義

次のような X を見つける問題

$$\begin{array}{ll} \text{maximize} & \sum_{e \in X} w(e) \\ \text{subject to} & X \in \mathcal{F} \end{array}$$

最適解 (optimal solution) とは？

上の問題の最適解とは，$X^* \in \mathcal{F}$ で，次を満たすもののこと

$$\text{任意の } X \in \mathcal{F} \text{ に対して，} \sum_{e \in X^*} w(e) \geq \sum_{e \in X} w(e)$$

つまり，組合せ最適化では，最適解を見つけたい

組合せ最適化問題：用語 (最適値)

組合せ最適化問題：定義

次のような X を見つける問題

$$\begin{array}{ll} \text{maximize} & \sum_{e \in X} w(e) \\ \text{subject to} & X \in \mathcal{F} \end{array}$$

最適値 (optimal value) とは？

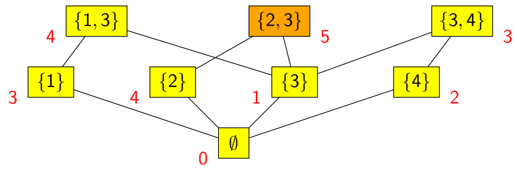
上の問題の最適値とは，最適解の目的関数値

注：「最適解」と「最適値」は異なる概念

例 1：ナップサック問題 — 最適解と最適値

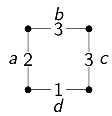
商品	1	2	3	4
収入 [万円]	3	4	1	2
重さ [kg]	2	3	1	3

赤字が収入 (つまり目的関数値) を表す

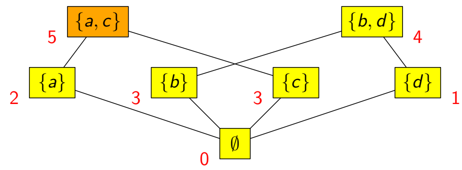


$\{2,3\}$ は最適解であり，最適値は 5

例 2：最大重みマッチング問題 — 最適解と最適値



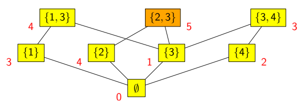
赤字が収入 (つまり目的関数値) を表す

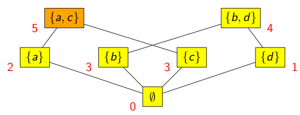


$\{a, c\}$ は最適解であり，最適値は 5

目次

問：2 つの例に共通することは？





どちらの例でも次の性質が成り立っている

$$X \in \mathcal{F} \text{ かつ } Y \subseteq X \quad \text{ならば } Y \in \mathcal{F}$$

(日本語訳：X が許容解であるならば，その部分集合 Y も許容解である)

この性質を持つ $\mathcal{F}$ が組合せ最適化には頻出する

- ということなので，名前を付ける

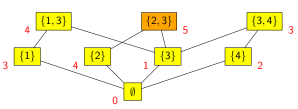
独立集合族：定義

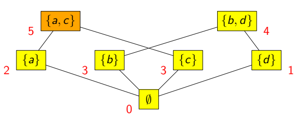
非空な有限集合 E，有限集合族 $\mathcal{F} \subseteq 2^E$

独立集合族 (independence system) とは？

$\mathcal{F}$ が E 上の独立集合族であるとは，以下の 2 つを満たすこと

1. $\emptyset \in \mathcal{F}$
2. $X \in \mathcal{F}$ かつ $Y \subseteq X$ ならば，$Y \in \mathcal{F}$





抽象単体複体 (abstract simplicial complex) と呼ばれることもある

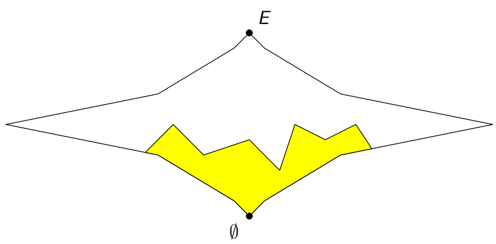
独立集合族：イメージ

非空な有限集合 E，有限集合族 $\mathcal{F} \subseteq 2^E$

独立集合族 (independence system) とは？

$\mathcal{F}$ が E 上の独立集合族であるとは，以下の 2 つを満たすこと

1. $\emptyset \in \mathcal{F}$
2. $X \in \mathcal{F}$ かつ $Y \subseteq X$ ならば，$Y \in \mathcal{F}$



目次

独立集合族とアルゴリズム

非空な有限集合 E，独立集合族 $\mathcal{F} \subseteq 2^E$

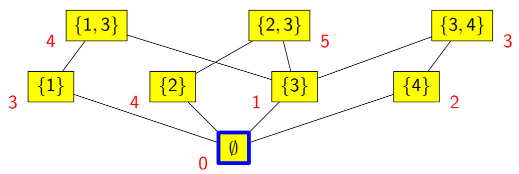
解くべき問題

$$\begin{array}{ll} \text{maximize} & \sum_{e \in X} w(e) \\ \text{subject to} & X \in \mathcal{F} \end{array}$$

考えやすいアルゴリズム (の 1 つ)

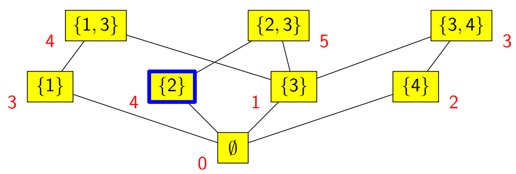
- 貪欲アルゴリズム (greedy algorithm)

貪欲アルゴリズム：例 (1/4)



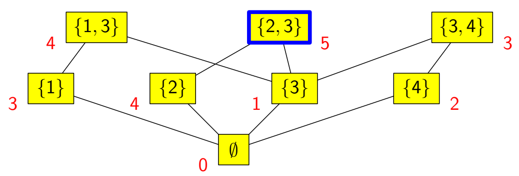
まず，$\emptyset$ からスタート

貪欲アルゴリズム：例 (2/4)



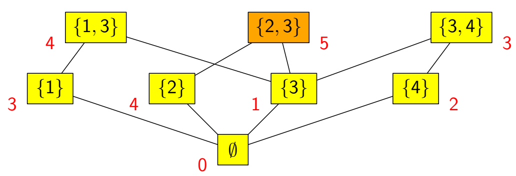
目的関数値が最も大きくなる「一歩」を踏み出す

貪欲アルゴリズム：例 (3/4)



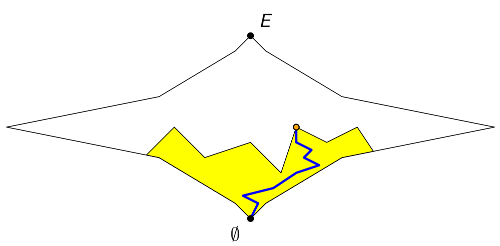
目的関数値が最も大きくなる「一歩」を踏み出す

貪欲アルゴリズム：例 (4/4)



それ以上進めなくなったら終了

貪欲アルゴリズム：イメージ



貪欲アルゴリズム：疑似コード

非空な有限集合 E，独立集合族 $\mathcal{F} \subseteq 2^E$

貪欲アルゴリズム

1. $X \leftarrow \emptyset$
2. 次を満たす $x \in E - X$ を見つける
   - $X \cup \{x\} \in \mathcal{F}$
   - $X \cup \{y\} \in \mathcal{F}$ を満たす任意の $y \in E - X$ に対して，
   $$w(x) \geq w(y)$$
3. そのような x が無ければ，X を出力し，終了
4. あれば，$X \leftarrow X \cup \{x\}$ として，2 に戻る

マトロイドと貪欲アルゴリズム

非空な有限集合 E，独立集合族 $\mathcal{F} \subseteq 2^E$

定理

次の 2 つは同値

1. 任意の重み関数 $w\colon E \to \mathbb{R}_+$ に対して，貪欲アルゴリズムが最適解を出力する
2. 独立集合族 $\mathcal{F}$ がマトロイドである

つまり，

- マトロイド (matroid) とは特殊な性質を持つ独立集合族
- マトロイドに対しては，貪欲アルゴリズムが必ず最適解を出力する

⇝ マトロイドはとても性質のよい独立集合族

目次

テーマ：解きやすい組合せ最適化問題が持つ「共通の性質」

疑問

どうしてそのような違いが生まれるのか？

⇝ 解きやすい問題が持つ「共通の性質」は何か？

回答

よく分かっていない

しかし，部分的な回答はある

部分的な回答

問題が「マトロイド的構造」を持つと解きやすい

ポイント

効率的アルゴリズムが設計できる背景に「美しい数理構造」がある

この講義では，その一端に触れたい

今日の重要概念

- 組合せ最適化問題
  - 台集合，許容集合，許容解，目的関数，最適解，最適値
- 独立集合族 (とそのイメージ)
- 貪欲アルゴリズム
- マトロイドは (まだ定義してないけど) とてもよい独立集合族であるということ

次回の予告

次回の予告

- マトロイドの定義と例

次回以降，前半の流れ (第 7 回まで)

- マトロイドと最小全域木問題の関係
  (貪欲アルゴリズム = Kruskal のアルゴリズム)
- 定理の証明

ここまでが，マトロイドの基礎

後半 (第 8 回以降)

- 「マトロイド交わり定理」の証明とアルゴリズム

これは二部グラフの最大マッチングなどに関係

残った時間の使い方

- 演習問題をやる
  - 相談推奨 (ひとりでやらない)
- 質問をする
  - 教員は巡回
- 退室時，小さな紙に感想など書いて提出する ← 重要
  - 内容は何でも OK
  - 匿名で OK