予定

- 10:30—11:20 講義
  - 劣モジュラ関数の定義，最大化問題の分類，例
- 11:40—12:30 講義
  - 研究の歴史，基本的な近似技法
- 14:00—15:00 講義
  - 最近の近似技法
- 15:15—16:15 演習
- 16:30—17:30 講義
  - これまでの研究成果の紹介

近似に使われる技法

- 貪欲算法(greedy algorithm)
- 局所探索(local search)
- 部分列挙, 推測(partial enumeration, guess)
- 乱択化(randomization)
- 連続緩和＋丸め(continuous relaxation＋rounding)

- ラグランジュ緩和 (Lagrangian relaxation)

14:00–15:00の予定

- 最近の近似アルゴリズムで用いられるテクニック
  - 連続緩和と連続的貪欲算法
  - パイプ丸め

連続緩和と丸め

1つのマトロイド制約下での
単調劣モジュラ関数最大化に対する
Vondrákの (1-1/e)近似アルゴリズムを例として

マトロイド制約下での単調劣モジュラ関数最大化

Maximize $f(X)$ **subject to** $X \in \mathcal{I}$

$f : 2^N \to \mathbf{R}_+$, 単調劣モジュラ

$\mathcal{I} \subseteq 2^N$, マトロイド(の独立集合族)

Vondrák(2008)の近似アルゴリズム: (1-1/e)近似
(continuous greedy + pipage rounding)

マトロイドの定義

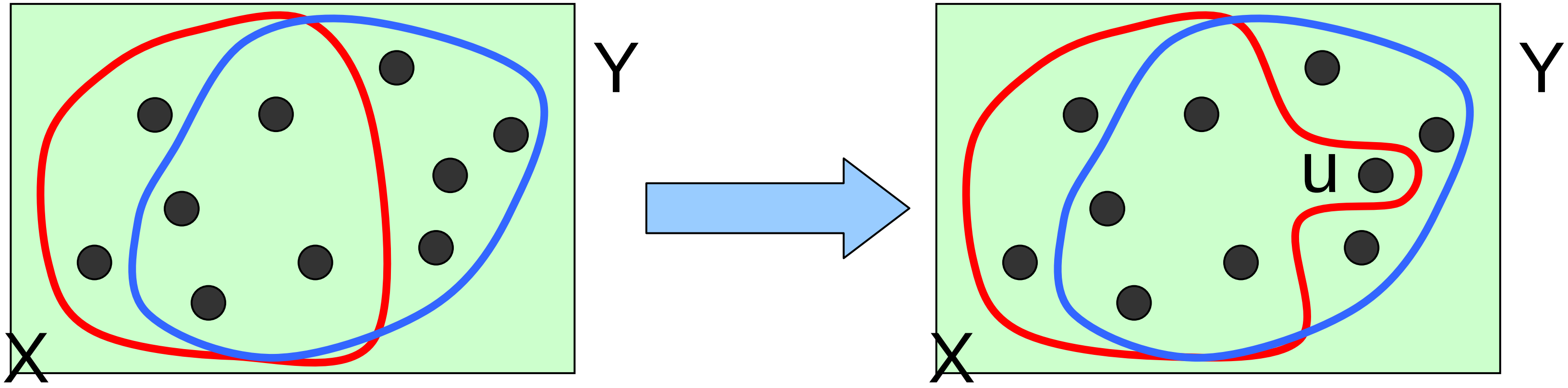
マトロイドの定義

定義：$\mathcal{I} \subseteq 2^N$ はマトロイド（の独立集合族）

$\Updownarrow$

(i) $Y \in \mathcal{I},\ X \subseteq Y \Rightarrow X \in \mathcal{I}$

(ii) $X, Y \in \mathcal{I}$, $|X| < |Y| \Rightarrow \exists u \in Y - X : X + u \in \mathcal{I}$



$X \in \mathcal{I}$： マトロイド $\mathcal{I}$ の独立集合

マトロイドの例

- 一様マトロイド: 整数 k に対し,

$$\mathcal{I} = \{X \subseteq N \mid |X| \leq k\}$$

- 分割マトロイド:

N の分割 $\{N_1, N_2, \ldots, N_t\}$, 正の整数 $k_1, k_2, \ldots, k_t$ に対し

$$\mathcal{I} = \{X \subseteq N \mid |X \cap N_i| \leq k_i \ (i = 1, 2, \ldots, t)\}$$

- グラフ的マトロイド: 無向グラフ G=(V, E) に対し

$$\mathcal{I} = \{X \subseteq E \mid X \text{ は } G \text{ の森}\}$$

- 行列的マトロイド: 行列 A = $[a_1, a_2, \ldots, a_n]$ に対し

$$\mathcal{I} = \{X \subseteq N \mid \{a_i | i \in X\} \text{ は一次独立}\}$$

演習

マトロイドの基

定義：$X \in \mathcal{I}$ はマトロイド$\mathcal{I}$の基 (base)

$$\iff \forall u \in N - X,\ \ X + u \notin \mathcal{I}$$

定理: X, Y: マトロイドの基➔|X| = |Y|

演習

マトロイド$\mathcal{I}$ の基の族

$$\mathcal{B} \equiv \{X \subseteq N \mid X \in \mathcal{I},\ \ X \text{は基}\}$$

- 一様マトロイドの基の族:

$$\mathcal{B} = \{X \subseteq N \mid |X| = k\}$$

- 分割マトロイドの基の族:

$$\mathcal{B} = \{X \subseteq N \mid |X \cap N_i| = k_i \ (i = 1, 2, \ldots, t)\}$$

マトロイド制約下での単調劣モジュラ関数最大化

Maximize $f(X)$ subject to $X \in \mathcal{I}$

$f : 2^N \to \mathbf{R}_+$, 単調劣モジュラ

$\mathcal{I} \subseteq 2^N$, マトロイド(の独立集合族)

Xはマトロイドの基と仮定しても良い

目的関数は単調➔マトロイドの基の中に必ず最適解が存在

Vondrák(2008)の近似アルゴリズム: (1-1/e)近似
(continuous greedy + pipage rounding)

仮定: X⊆N に対して,「f(X)の値の計算」「X∈$\mathcal{I}$ の判定」
が単位時間で出来る
(f の関数値評価オラクルと $\mathcal{I}$ のメンバーシップオラクル
が与えられている)

Vondrákの(1−1/e)近似アルゴリズム

アルゴリズムの手順:

① 元問題を$\{0,1\}^n$上での離散最適化問題と見なす

② $[0,1]^n$上での制約付き連続最適化(最大化)問題へ緩和

③ 連続的貪欲算法(continuous greedy) により,
緩和問題の(1-1/e)近似解を求める ➔ 緩和解 $x^* \in [0,1]^n$

④ パイプ丸め(pipage rounding) により, 緩和解 x^* を
$f(X^*) \geqq F(x^*)$ を満たす元問題の許容解 $X^* \subseteq N$ に丸める

$X^* \subseteq N$ は元問題の (1-1/e)近似解

$\because\ f(X^*) \geqq F(x^*)$
$\geqq (1-1/e) \times$(連続緩和の最適値)
$\geqq (1-1/e) \times$(元問題の最適値)

集合と{0, 1}ベクトルの対応

アルゴリズムの手順:
① 元問題を$\{0,1\}^n$上での離散最適化問題と見なす

N の部分集合 X を{0,1}ベクトルと見なす

例: N = {1,2,3,4,5}, X = {2,4,5} ➔ $\chi_X = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

劣モジュラ関数 $f: 2^N \to R$ --- $\{0,1\}^n$ 上で定義された関数

マトロイド $\mathcal{I} \subseteq 2^N$ --- $\{0,1\}^n$の部分集合

連続最適化への緩和

アルゴリズムの手順:
② $[0,1]^n$上での制約付き連続最適化(最大化)問題へ緩和

劣モジュラ関数 f: $2^N \rightarrow R$ --- $\{0,1\}^n$ 上で定義された関数
➔ $[0,1]^n$ 上で定義された関数 F: $[0,1]^n \rightarrow R$ に置き換える

マトロイド $\mathcal{I} \subseteq 2^N$ --- $\{0,1\}^n$の部分集合
➔ $[0,1]^n$ に含まれる多面体 $P(\mathcal{I})$ に置き換える

得られた緩和問題

Maximize F(x) subject to $x \in P(\mathcal{I})$

目的関数の緩和:多重線形拡張

- 集合関数$f: 2^N \rightarrow R$ の多重線形拡張(multilinear extension)

  $F: [0,1]^n \rightarrow R$

$$F(x) = \sum_{X \in 2^N} \Big[\prod_{j \in X} x(j)\Big]\Big[\prod_{j \in N \setminus X} (1 - x(j))\Big] f(X)$$

演習

$$\left.\begin{array}{ll} \text{確率 } x(j) \text{ で} & j \in X \\ \text{確率 } 1 - x(j) \text{ で} & j \notin X \end{array}\right\} \text{としたときの } f(X) \text{ の期待値}$$

例: N = {1, 2}, x_1 = 1/4, x_2 = 1/3

➔ F(x) = 3/4*2/3 f(∅) +1/4*2/3 f({1})

+ 3/4*1/3 f({2}) + 1/4*1/3 f({1,2})

多重線形拡張の性質

集合関数f:2^N→R の多重線形拡張 F: $[0,1]^n$→R は次を満たす

F は f の拡張 $(F(\chi_X) = f(X)\ (\forall X \in 2^N))$

F(x)の近似値はランダムサンプリングにより高い確率で計算可能

※Chernoff bound を利用

多重線形拡張の性質

とくに，$f: 2^N \to R$ が単調劣モジュラのとき

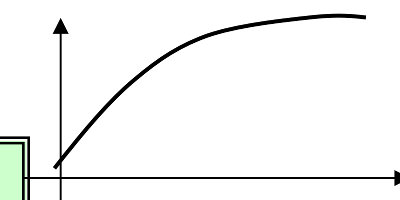
F は 単調非減少（$x \leqq y \Rightarrow F(x) \leqq F(y)$）

F は劣モジュラ（$F(x)+F(y) \geqq F(x \vee y)+F(x \wedge y)$ ）

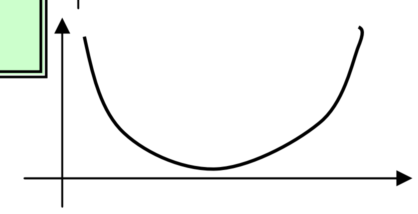
演習

$(x \vee y)_i = \max(x_i, y_i)$, $(x \wedge y)_i = \min(x_i, y_i)$

$d \geqq 0$ に対し, $F_d^y(t) = F(y+td)$ は(t に関する)凹関数

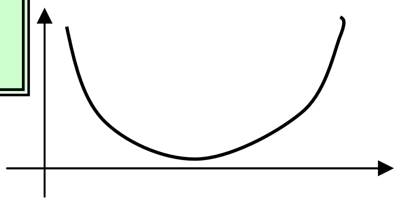


$F_{ij}^y(t) = F(y + t(\chi_i - \chi_j))$ は(t に関する)凸関数



多重線形拡張の凸性の証明

$F_{ij}^{y}(t) = F(y + t(\chi_i - \chi_j))$ は(t に関する)凸関数



$p_x(S) = \Big[\prod_{j \in S} x(j)\Big]\Big[\prod_{j \in N \setminus (S+i+j)} (1 - x(j))\Big]$ とおくと

$$F(x) = \sum_{S \subseteq N \setminus \{i,j\}} p_x(S)\Big[(1 - x_i)(1 - x_j) f(S) + (1 - x_i) x_j f(S + j) + x_i (1 - x_j) f(S + i) + x_i x_j f(S + i + j)\Big]$$

と書ける

よって $F_{ij}^{y}(t)$ は t に関する2次関数

2次項の係数は

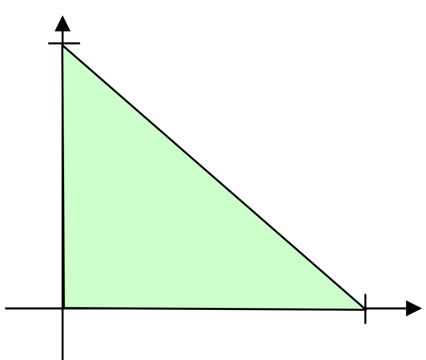
$$\sum_{S \subseteq N \setminus \{i,j\}} p_x(S)[-f(S) + f(S + j) + f(S + i) - f(S + i + j)] \geq 0$$

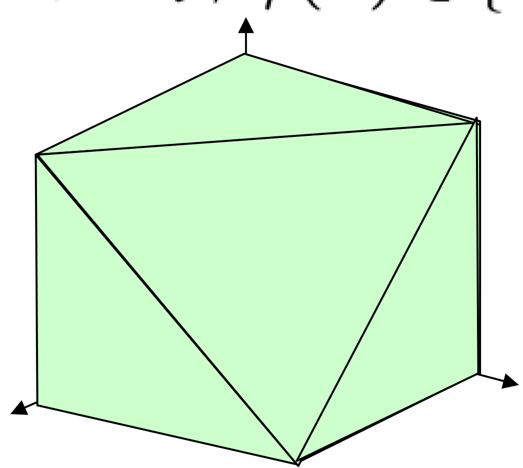
制約の連続緩和:マトロイド多面体

マトロイド $\mathcal{I}$ の独立集合 X を$\{0,1\}$ベクトル χ_X と対応づける

例: N = {1,2,3,4,5}, X = {2,4,5} ➔ $\chi_X = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

マトロイド $\mathcal{I}$ に対し，$\mathcal{I}$のマトロイド多面体 $P(\mathcal{I})$
$= \{0,1\}$ベクトル集合 $\{\chi_X \in \{0,1\}^N \mid X \in \mathcal{I}\}$ の凸包
$= \{x \in [0,1]^N \mid \sum_{i \in S} x_i \le \rho(S) \ \ (S \subseteq N)\}$
$(\exists \rho : 2^N \to \mathbf{R}$, 単調劣モジュラ, $\rho(S) \in \{0,1,\ldots,n\})$





$\mathcal{I}$ のメンバーシップオラクルが利用可
➔ ρ の値は容易に計算可能

制約の連続緩和:基多面体

マトロイド $\mathcal{I}$ に対し，$\mathcal{I}$のマトロイド多面体$P(\mathcal{I})$

$=\{0,1\}$ベクトル集合$\{\chi_X \in \{0,1\}^N \mid X \in \mathcal{I}\}$ の凸包

$=\{x \in [0,1]^N \mid \sum_{i\in S} x_i \le \rho(S) \ \ (S \subseteq N)\}$

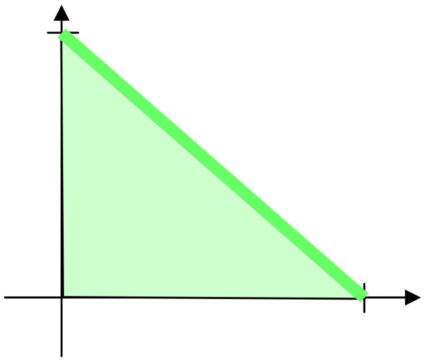
$(\exists \rho : 2^N \to \mathbf{R}$, 単調劣モジュラ, $\rho(S) \in \{0,1,\ldots,n\})$

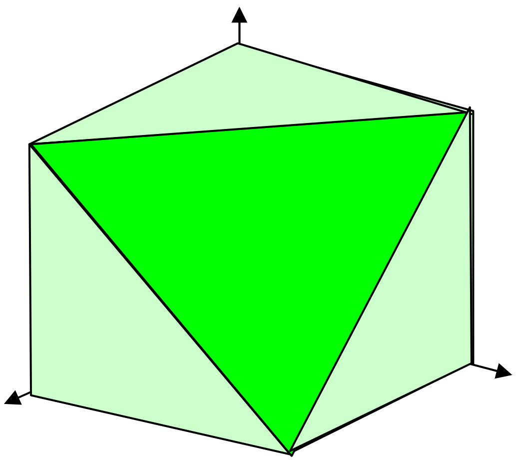
$\mathcal{I}$の基多面体$B(\mathcal{I})$

$=$ マトロイド $\mathcal{I}$ の基に対応する$\{0,1\}$ベクトル集合

$\{\chi_X \in \{0,1\}^N \mid X \in \mathcal{B}\}$ の凸包

$=\{x \in P(\mathcal{I}) \mid \sum_{i\in N} x_i = \rho(N)\}$





マトロイド多面体の例

- 一様マトロイドの独立集合族と基の族:

$$\mathcal{I} = \{X \subseteq N \mid |X| \leq k\} \qquad \mathcal{B} = \{X \subseteq N \mid |X| = k\}$$

一様マトロイドのマトロイド多面体と基多面体:

$$P(\mathcal{I}) = \{x \in [0,1]^n \mid \textstyle\sum_{i=1}^{n} x_i \leq k\}$$

$$B(\mathcal{I}) = \{x \in [0,1]^n \mid \textstyle\sum_{i=1}^{n} x_i = k\}$$

- 分割マトロイドの独立集合族と基の族

$$\mathcal{I} = \{X \subseteq N \mid |X \cap N_i| \leq k_i \ (i = 1, 2, \ldots, t)\}$$

$$\mathcal{B} = \{X \subseteq N \mid |X \cap N_i| = k_i \ (i = 1, 2, \ldots, t)\}$$

分割マトロイドのマトロイド多面体と基多面体:

$$P(\mathcal{I}) = \{x \in [0,1]^n \mid \textstyle\sum_{j \in N_i} x_j \leq k_i \ (i = 1, 2, \ldots, t)\}$$

$$B(\mathcal{I}) = \{x \in [0,1]^n \mid \textstyle\sum_{j \in N_i} x_j = k_i \ (i = 1, 2, \ldots, t)\}$$

連続的貪欲算法

アルゴリズムの手順:

③ 連続的貪欲算法(continuous greedy algorithm) により,
緩和問題の(1-1/e)近似解を求める ➔ 緩和解 $x^* \in [0,1]^n$

連続的貪欲算法(k=十分大きな正整数, $\delta = 1/k$)

Step 0: x := (0,0,...,0)
Step 1: 反復回数が k 回になったら終了, x* = x を出力
Step 2: $\nabla F(x)$を計算
Step 3: $\max\{\nabla F(x)^T y \mid y \in B(\mathcal{I})\}$の最適解 y* を求める
Step 4: $x := x + \delta y^*$
Step 5: Step 1 に戻る

連続的貪欲算法の解の性質

連続的貪欲算法（k＝十分大きな正整数, δ＝1/k）

Step 0: x := (0,0,...,0)
Step 1: 反復回数が k 回になったら終了, x*＝x を出力
Step 2: ∇F(x)を計算
Step 3: $\max\{\nabla F(x)^T y \mid y \in B(\mathcal{I})\}$の最適解 y* を求める
Step 4: x := x + δ y*
Step 5: Step 1 に戻る

補題：最終的に得られた $x^* \in B(\mathcal{I})$

∵ y_i^*: 第 i 反復での y*

➔ $x^* = \delta(y_1^* + y_2^* + \ldots + y_k^*) = (1/k)(y_1^* + y_2^* + \ldots + y_k^*)$

➔ x* は $B(\mathcal{I})$ の中のベクトルの凸結合

連続的貪欲算法の近似比の解析

連続的貪欲算法（k＝十分大きな正整数, δ＝1/k）

Step 0: x := (0,0,...,0)
Step 1: 反復回数が k 回になったら終了, x*＝x を出力
Step 2: ∇F(x)を計算
Step 3: max{ $\nabla F(x)^T y$ | y∈$B(\mathcal{I})$}の最適解 y* を求める
Step 4: x := x + δ y*
Step 5: Step 1 に戻る

Key Lemma: Step 3 において

$$\nabla F(x)^T y^* \geqq \mathrm{ROPT} - F(x)$$

（ROPT＝緩和問題の最適値）

連続的貪欲算法の近似比の解析

Key Lemma: Step 3 において

$$\nabla F(x)^T y^* \geqq \mathrm{ROPT} - F(x)$$

（ROPT＝緩和問題の最適値）

⇒ $F(x+\delta y^*) - F(x) \fallingdotseq \nabla F(x)^T(\delta y^*)$
$\geqq \delta(\mathrm{ROPT} - F(x))$

⇒ $\mathrm{ROPT} - F(x+\delta y^*) \leqq (1-\delta)(\mathrm{ROPT} - F(x))$

⇒ $\mathrm{ROPT} - F(x^*) \leqq (1-\delta)^{1/\delta}(\mathrm{ROPT} - F(0))$
$\leqq (1/e)\mathrm{ROPT}$

⇒ $F(x^*) \geqq (1 - 1/e)\mathrm{ROPT}$

連続的貪欲算法の問題点と修正

連続的貪欲算法（k＝十分大きな正整数，$\delta=1/k$）

Step 0: $x := (0,0,\ldots,0)$
Step 1: 反復回数が k 回になったら終了，$x^*=x$ を出力
Step 2: $\nabla F(x)$を計算
Step 3: $\max\{\nabla F(x)^T y \mid y \in \mathcal{B}(\mathcal{I})\}$の最適解 y^* を求める
Step 4: $x := x + \delta y^*$
Step 5: Step 1 に戻る

反復回数 k を小さくしたい（多項式回）

➔ $\delta=1/k$ が大きくなる

➔ $F(x+\delta y^*) - F(x) \fallingdotseq \nabla F(x)^T(\delta y^*)$ が成り立たない

➔ (1-1/e)近似が導けない

解決策: $\nabla F(x)$ の代わりに類似した値を利用

➔ 多項式時間で(1-1/e)近似が可能

パイプ丸め

アルゴリズムの手順:

④ パイプ丸め(pipage rounding) により，緩和解 x^* を $f(X^*) \geqq F(x^*)$ を満たす元問題の許容解 $X^* \subseteq N$ に丸める

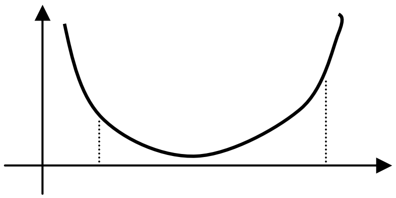
基本となるアイディア

y_i, y_j が非整数➔y_iを増やし，y_jを減らす（またはその逆）

$$t^- = \min\{t | y + t(\chi_i - \chi_j) \in B(\mathcal{I})\}, \quad x^- = y + t^-(\chi_i - \chi_j)$$

$$t^+ = \max\{t | y + t(\chi_i - \chi_j) \in B(\mathcal{I})\}, \quad x^+ = y + t^+(\chi_i - \chi_j)$$

⇨ $F(x^-) \geqq F(x)$ または $F(x^+) \geqq F(x)$ の どちらか一方は必ず成立



$F_{ij}^{y}(t) = F(y + t(\chi_i - \chi_j))$ は(t に関する)凸関数

パイプ丸めの例

例：分割マトロイド

$N=\{1,2,\ldots,6\}$, $N_1=\{1,2,3\}$, $N_2=\{4,5,6\}$

制約 $x_1+x_2+x_3=1$, $x_4+x_5+x_6=1$, $0 \leqq x_j \leqq 1 (\forall j)$

制約を満たしつつ，かつ関数値を減らさないように，2つの成分の増減を繰り返す

(0.5, 0.2, 0.3, 0.3, 0.4, 0.3)

非整数成分に注目

0.5+0.2+0.3=1 (整数)➔非整数成分がもう一つ存在➔0.2

0.5 を増やし 0.2 を減らす，　または　0.5 を減らし 0.2 を増やす

パイプ丸めの例

例：分割マトロイド

$N=\{1,2,\ldots,6\}$, $N_1=\{1,2,3\}$, $N_2=\{4,5,6\}$

制約 $x_1+x_2+x_3=1$, $x_4+x_5+x_6=1$, $0 \leqq x_j \leqq 1 (\forall j)$

制約を満たしつつ，かつ関数値を減らさないように，2つの成分の増減を繰り返す

(0.5, 0.2, 0.3, 0.3, 0.4, 0.3)

(0.0, 0.7, 0.3, 0.3, 0.4, 0.3)

(0.0, 1.0, 0.0, 0.3, 0.4, 0.3)

(0.0, 1.0, 0.0, 0.7, 0.0, 0.3)

(0.0, 1.0, 0.0, 1.0, 0.0, 0.0)

一般のマトロイドの場合はもっと複雑

最終的に{0,1}ベクトルが得られる

パイプ丸めのアルゴリズム

演習

入力： $y_0 \in [0,1]^n$

出力： $y^* \in \{0,1\}^n$, $F(y^*) \geqq F(y_0)$

Step 0: $y := y_0$

Step 1: yが{0,1}ベクトルならば終了． $y^* = y$ を出力

Step 2: 2つの非整数成分 y_i, y_j を含み，かつ $y(S) = \rho(S)$を満たす極小な $S \subseteq N$ を求める（必ず存在）

Step 3:

$$t^- = \min\{t | y + t(\chi_i - \chi_j) \in B(\mathcal{I})\},\ x^- = y + t^-(\chi_i - \chi_j)$$

$$t^+ = \max\{t | y + t(\chi_i - \chi_j) \in B(\mathcal{I})\},\ x^+ = y + t^+(\chi_i - \chi_j)$$

（必ず $t^- > 0$, $t^+ < 0$成立）

Step 4: x^- と x^+ のうち，F の関数値の大きい方を y とおく

Step 5: Step 1 に戻る

補題: このアルゴリズムの反復回数≦n^2

Vondrákの(1−1/e)近似アルゴリズム

アルゴリズムの手順:

① 元問題を$\{0,1\}^n$上での離散最適化問題と見なす

② $[0,1]^n$上での制約付き連続最適化(最大化)問題へ緩和

③ 連続的貪欲算法(continuous greedy) により,
緩和問題の(1-1/e)近似解を求める ➔ 緩和解 $x^* \in [0,1]^n$

④ パイプ丸め(pipage rounding) により, 緩和解 x^* を
$f(X^*) \geqq F(x^*)$ を満たす元問題の許容解 $X^* \subseteq N$ に丸める

定理: $X^* \subseteq N$ は元問題の (1-1/e)近似解

演習問題(その3)

(1) p.3-8: 4つの例がマトロイドであることを証明せよ.

(2) p.3-9: 定理を証明せよ.

(3) p.3-14: 多重線形拡張の勾配ベクトルを計算せよ.

(4) p.3-16: F の単調性, 劣モジュラ性, 凹性を証明せよ.

(5) p.3-29: パイプ丸めのStep 3 にて$t^+>0$, $t^-<0$ が成り立つことを示せ.