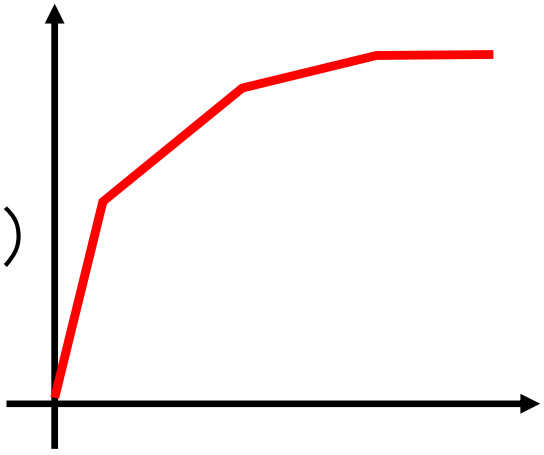
数理経済学
第7回
資源配分問題

塩浦昭義
東京工業大学 経営工学系
shioura.a.aa@m.titech.ac.jp

単純な資源配分問題

単純な資源配分問題：定式化



- n 人の子供に r 個のリンゴを配分
  - 配分単位は整数（リンゴを切り分けることは不可）
- 全員の満足度の合計値を最大にしたい
  - 1個あたりの満足度は単調非増加
    （1個目の嬉しさ≧2個目の嬉しさ≧3個目の嬉しさ≧．．．）

定式化：

最大化 $\sum_{i=1}^{n} g_i(x_i)$　条件 $\sum_{i=1}^{n} x_i = r,\ x_i \geq 0$, 整数 $(\forall i = 1, \ldots, n)$

$g_i(t)$: 子供 i の満足度関数

仮定: 差分 $g_i(t+1) - g_i(t)$ は t に関して単調非増加

(離散)凹関数
とよぶ

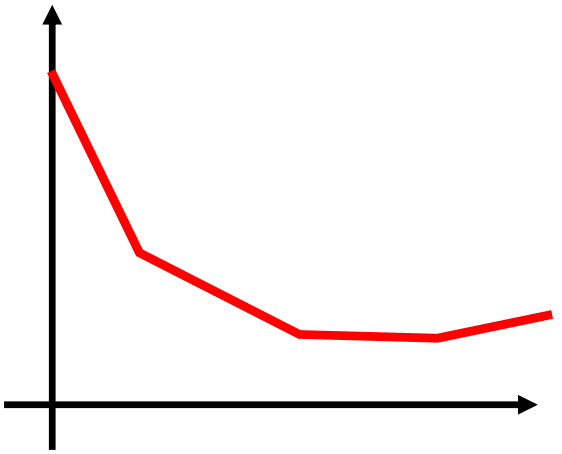
単純な資源配分問題：最小化版

- 目的関数を −1 倍すると最小化問題

最小化 $\sum_{i=1}^{n} f_i(x_i)$　条件 $\sum_{i=1}^{n} x_i = r,\ x_i \geq 0$, 整数 $(\forall i = 1, \ldots, n)$

$f_i(t)$: 差分 $f_i(t+1) - f_i(t)$ は t に関して単調非減少

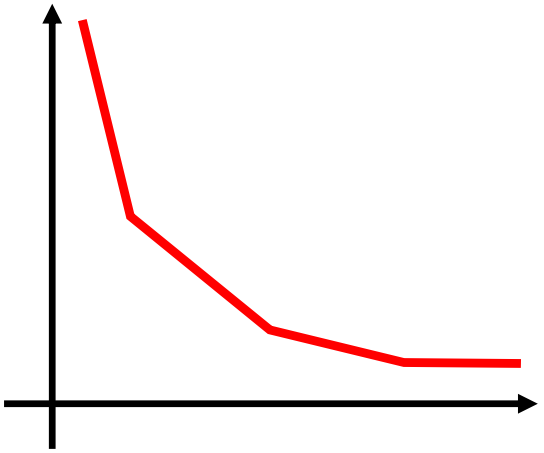
(離散)凸関数
とよぶ



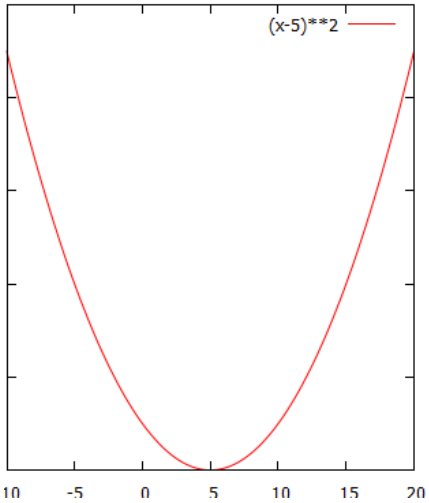
単純な資源配分問題の例1

- n 個の作業（ジョブ）に労働資源（労働者，CPU，材料など）を配分
- 処理時間の和を最小化したい
  - 作業 i の処理時間 $f_i(x_i)$ は離散凸
    （ひとつの作業に大勢の人が携わっても効率は上がらない）

最小化 $\sum_{i=1}^{n} f_i(x_i)$　条件 $\sum_{i=1}^{n} x_i = r,\ x_i \geq 0$, 整数 $(\forall i = 1, \ldots, n)$



単純な資源配分問題の例2



議員定数配分問題

- 国会などの議員総数→各選挙区に配分
- 選挙区数= n, 議員定数= r
- 第 i 選挙区の人口 p(i) に比例した議席数を配分したい
  - 理想：選挙区の人口に比例した議席数 $r \times p_i / \sum_j p_j$
  - 理想値の実現は無理➔なるべく近い値にする

$\sum_{i=1}^{n} \left(x_i - \frac{p_i}{\sum_j p_j} r \right)^2$ としては駄目

最小化 $\sum_{i=1}^{n} \left(\frac{x_i}{p_i} - \frac{r}{\sum_j p_j} \right)^2$

条件 $\sum_{i=1}^{n} x_i = r,\ x_i \geq 0$, 整数 $(\forall i = 1, \ldots, n)$

目的関数を $\sum_{i=1}^{n} \left| \frac{x_i}{p_i} - \frac{r}{\sum_j p_j} \right|$ に変更しても可

（結果は変わる可能性あり）

単純な資源配分問題のアルゴリズム

貪欲アルゴリズム

※以降では，最大化版の問題を考える

貪欲アルゴリズム：目的関数値が最も増えるように，追加配分する

ステップ0：$(x_1, \ldots, x_n) = (0, \ldots, 0)$とする．
ステップ1：$\sum_{i=1}^{n} x_i = r$ ➔ $(x_1, \ldots, x_n)$ を出力して終了．
ステップ2：$g_i(x_i + 1) - g_i(x_i)$ が最大の i に対し x_i を1増やす．
ステップ1に戻る．

貪欲アルゴリズム：実行例1

最大化　$-(2x_1-7)^2-(3x_2-10)^2-(x_3-8)^2$
条件　　$x_1+x_2+x_3=6, x_i \geq 0$, 整数

表を最初に書くとわかりやすい（手計算の場合）

	1	2	3
$g_i(0)$	-49	-100	-64
$g_i(1)$	-25	-49	-49
$g_i(2)$	-9	-16	-36
$g_i(3)$	-1	-1	-25
$g_i(4)$	-1	-4	-16

	1	2	3
$g_i(1)-g_i(0)$	24	51	15
$g_i(2)-g_i(1)$	16	33	13
$g_i(3)-g_i(2)$	8	15	11
$g_i(4)-g_i(3)$	0	-3	9

x2→x2→x1→x1→x2→x3
の順に1ずつ増やしていく

貪欲アルゴリズム：実行例2

最大化　$-(2x_1-1)^2-(3x_2-2)^2-3x_3^2$
条件　　$x_1+x_2+x_3=6, x_i \geq 0$, 整数

表を最初に書くとわかりやすい（手計算の場合）

	1	2	3
$g_i(0)$	-1	-4	0
$g_i(1)$	-1	-1	-3
$g_i(2)$	-9	-16	-12
$g_i(3)$	-25	-49	-27
$g_i(4)$	-49	-100	-48

	1	2	3
$g_i(1)-g_i(0)$	0	3	-3
$g_i(2)-g_i(1)$	-8	-15	-9
$g_i(3)-g_i(2)$	-16	-33	-15
$g_i(4)-g_i(3)$	-24	-51	-21

x2→x1→x3→x1→x3→x2 or x3
の順に1ずつ増やしていく

貪欲アルゴリズムの性質

正当性の証明に利用

アルゴリズムの動きより，以下が成り立つ

命題 アルゴリズムの終了時のベクトル $(x_1, \dots, x_n)$ を$(x_1^*, \dots, x_n^*)$ とおくと，次の条件(OPT)が成り立つ：
(OPT) $x_i^* > 0$ なる任意の i と任意の $j \neq i$ に対し，

$$g_i(x_i^*) - g_i(x_i^* - 1) \geq g_j(x_j^* + 1) - g_j(x_j^*)$$

証明：背理法．ある異なる $i, j \in \{1,2, \dots, n\}$ が存在して，$x_i^* > 0$ かつ

$$g_i(x_i^*) - g_i(x_i^* - 1) < g_j(x_j^* + 1) - g_j(x_j^*)$$

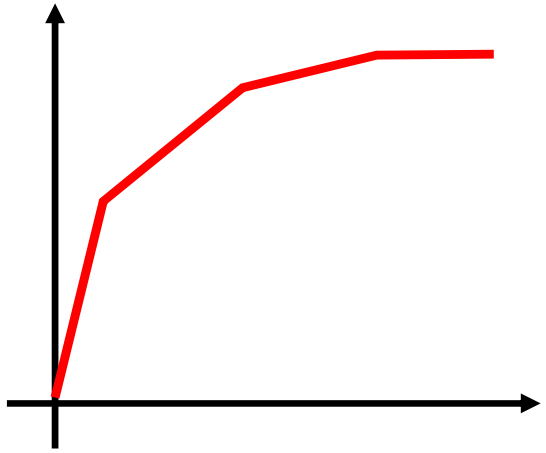
が成り立つと仮定．

アルゴリズムの動きより，変数 x_i の値を

$x_i^* - 1$ からx_i^* に増やす前に，変数 x_j の値をx_j^* から$x_j^* + 1$ に増やすべきだったことになり矛盾． ■

最適解の性質
および貪欲アルゴリズムの正当性

凹関数の性質



- **定義**: $g_i: \mathbb{Z} \to \mathbb{R}$ は(離散)凹関数

  ⇔ 整数 t に関して 差分 $g_i(t+1) - g_i(t)$ が単調非増加

命題 (離散)凹関数 $g_i: \mathbb{Z} \to \mathbb{R}$ および整数 α, β に対し,

$\alpha < \beta$ ➔ $g_i(\alpha) + g_i(\beta) \leq g_i(\alpha+1) + g_i(\beta-1)$

命題 (離散)凹関数 $g_i: \mathbb{Z} \to \mathbb{R}$ および整数 α に対し,

(i) $g_i(x_i)$ は $x_i = \alpha$ において最大

⇔ $g_i(\alpha-1) \leq g_i(\alpha)$ および $g_i(\alpha) \geq g_i(\alpha+1)$ が成立

(ii) $x_i \geq 0$ の条件下で, $g_i(x_i)$ は $x_i = 0$ において最大

⇔ $g_i(0) \geq g_i(1)$ が成立

証明: 演習問題(定義から簡単に導ける)

最適解の条件

定理 資源配分問題の実行可能解 $(x_1^*, \ldots, x_n^*)$ は最適解
←→ 次の条件(OPT)が成り立つ:
(OPT) $x_i^* > 0$ なる任意の i と任意の $j \neq i$ に対し,

$$g_i(x_i^*) - g_i(x_i^* - 1) \geq g_j\left(x_j^* + 1\right) - g_j\left(x_j^*\right)$$

証明は次のスライド.

この定理およびアルゴリズムの性質より,

アルゴリズムの正当性が直ちに導かれる

定理の対偶

定理 資源配分問題の実行可能解 $(x_1^*, \ldots, x_n^*)$ は最適解ではない
←→
$x_i^* > 0$ なる i および $j \neq i$ が存在して,

$$g_i(x_i^*) - g_i(x_i^* - 1) < g_j\left(x_j^* + 1\right) - g_j\left(x_j^*\right)$$

最適解の条件：必要性の証明

[➔の証明] こちらは（比較的）簡単.　対偶の「🡐」を証明する.

実行可能解 $(x_1^*, \dots, x_n^*)$ に対し,

$x_i^* > 0$ なるある i と, ある $j \neq i$ に対し

$$g_i(x_i^*) - g_i(x_i^* - 1) < g_j\left(x_j^* + 1\right) - g_j\left(x_j^*\right)$$

が成り立つと仮定

➔ x_i^* および x_j^* をそれぞれ $x_i^* - 1$ および $x_j^* + 1$ に置き換えた解は

実行可能解であり（$\because x_i^* > 0$）,

かつ目的関数がより大きい

$$\left(\because g_i(x_i^*) + g_j\left(x_j^*\right) < g_i(x_i^* - 1) + g_j\left(x_j^* + 1\right)\right)$$

➔ もとの解$(x_1^*, \dots, x_n^*)$ は最適解ではない　■

最適解の条件：十分性の証明

[←の証明]

方針：$(y_1^*, \dots, y_n^*)$: 最適解の中で$\sum_{i=1}^{n} |y_i^* - x_i^*|$ 最小とする.

このとき，$(y_1^*, \dots, y_n^*) = (x_1^*, \dots, x_n^*)$ が成り立つことを示せばよい.

$(y_1^*, \dots, y_n^*) \neq (x_1^*, \dots, x_n^*)$ と仮定，矛盾を導く.

$(y_1^*, \dots, y_n^*) \neq (x_1^*, \dots, x_n^*)$, $\sum_{i=1}^{n} y_i^* = \sum_{i=1}^{n} x_i^*$ なので，

ある j, k が存在して $y_j^* > x_j^*$, $y_k^* < x_k^*$ 成立.

凹関数の性質より

$$g_j(y_j^*) + g_j(x_j^*) \leq g_j(y_j^* - 1) + g_j(x_j^* + 1) \qquad \text{（式A）}$$

$$g_k(y_k^*) + g_k(x_k^*) \leq g_k(y_k^* + 1) + g_k(x_k^* - 1) \qquad \text{（式B）}$$

$x_k^* > y_k^* \geq 0$ なので，条件(OPT)より，

$$g_k(x_k^*) - g_k(x_k^* - 1) \geq g_j(x_j^* + 1) - g_j(x_j^*) \qquad \text{（式C）}$$

最適解の条件：十分性の証明（つづき）

ベクトル $(z_1^*, \dots, z_n^*)$ を次のように定義：

$$z_j^* = y_j^* - 1, \qquad z_k^* = y_k^* + 1, \qquad z_i^* = y_i^* \ (i \neq j, k)$$

$y_j^* > x_j^* \geq 0$ なので $(z_1^*, \dots, z_n^*)$ は実行可能解

$(z_1^*, \dots, z_n^*)$ の目的関数値 $-$ $(y_1^*, \dots, y_n^*)$の目的関数値

$$= g_j\left(y_j^* - 1\right) + g_k(y_k^* + 1) - g_j\left(y_j^*\right) - g_k(y_k^*)$$

式A,B,Cより

$$g_k(y_k^* + 1) - g_k(y_k^*) \geq g_k(x_k^*) - g_k(x_k^* - 1) \qquad \text{（式Bより）}$$

$$\geq g_j\left(x_j^* + 1\right) - g_j\left(x_j^*\right) \qquad \text{（式Cより）}$$

$$\geq g_j\left(y_j^*\right) - g_j\left(y_j^* - 1\right) \qquad \text{（式Aより）}$$

よって

$(z_1^*, \dots, z_n^*)$ の目的関数値 $-$ $(y_1^*, \dots, y_n^*)$の目的関数値$\geqq$0

最適解の条件：十分性の証明（つづき）

一方，$(y_1^*, \dots, y_n^*)$ は最適解であることから，

$(z_1^*, \dots, z_n^*)$ の目的関数値 $-$ $(y_1^*, \dots, y_n^*)$の目的関数値$\leqq 0$

$\therefore$ $(z_1^*, \dots, z_n^*)$ の目的関数値 $=$ $(y_1^*, \dots, y_n^*)$の目的関数値

よって $(z_1^*, \dots, z_n^*)$ も最適解．

しかし，$\sum_{i=1}^{n} |z_i^* - x_i^*| - \sum_{i=1}^{n} |y_i^* - x_i^*|$

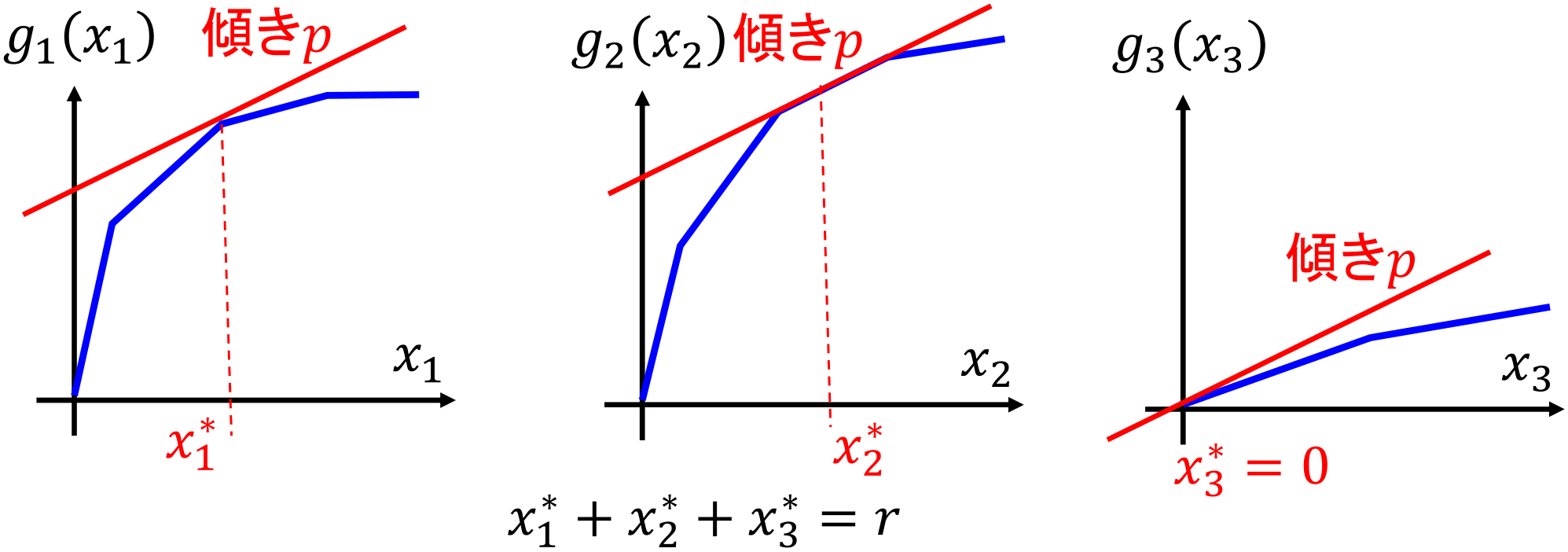
$= |z_j^* - x_j^*| + |z_k^* - x_k^*| - |y_j^* - x_j^*| - |y_k^* - x_k^*| = -2 < 0$

つまり，$\sum_{i=1}^{n} |z_i^* - x_i^*| < \sum_{i=1}^{n} |y_i^* - x_i^*|$ となり矛盾．

（証明終わり）

最適解の条件2

定理　資源配分問題の実行可能解$(x_1^*, x_2^*, \ldots, x_n^*)$に対し，
$(x_1^*, x_2^*, \ldots, x_n^*)$は最適解
⇔ 以下の条件を満たす実数pが存在：
(a) 任意の$x_i^* > 0$なる$i \in N$に対し，$g_i(x_i^*) - g_i(x_i^* - 1) \geq p$
(b) 任意の$j \in N$に対し，$g_j(x_j^* + 1) - g_j(x_j^*) \leq p$



最適解の条件2：必要性の証明

資源配分問題の実行可能解$(x_1^*, x_2^*, \ldots, x_n^*)$が最適解とする.

最適性の条件(OPT)が成立：
(OPT) $x_i^* > 0$ なる任意の i および任意の $j \neq i$ に対し,

$$g_i(x_i^*) - g_i(x_i^* - 1) \geq g_j\left(x_j^* + 1\right) - g_j\left(x_j^*\right)$$

$$p = \max_{j \in N}\left\{g_j\left(x_j^* + 1\right) - g_j\left(x_j^*\right)\right\}$$

とおくと，p の定義より (b) は直ちに成立.

$$g_k(x_k^* + 1) - g_k(x_k^*) = \max_{j \in N}\left\{g_j\left(x_j^* + 1\right) - g_j\left(x_j^*\right)\right\}(= p)$$

とすると，$x_i^* > 0$ なる任意の i に対し，条件 (OPT) より

$$g_i(x_i^*) - g_i(x_i^* - 1) \geq g_k(x_k^* + 1) - g_k(x_k^*) = p$$

よって (a) も成立. ■

最適解の条件2:十分性の証明

条件(a), (b) を満たす実行可能解$(x_1^*, x_2^*, \dots, x_n^*)$と
任意の実行可能解 $(y_1, y_2, \dots, y_n)$ に対し,

$$\sum_{i=1}^n g_i(x_i^*) \geq \sum_{i=1}^n g_i(y_i) \quad \text{（式A）}$$

を示せば良い.

関数 $g_i^p(x_i)$ を $g_i^p(x_i) = g_i(x_i) - px_i$ と定義.
すると, $\sum_{i=1}^n x_i^* = r = \sum_{i=1}^n y_i$ より, (式A)と次の不等式は等価:

$$\sum_{i=1}^n g_i^p(x_i^*) \geq \sum_{i=1}^n g_i^p(y_i) \quad \text{（式B）}$$

よって, (式B)を示せば良い.

(式B) は, 次の不等式から導かれる:

任意の $i \in N$ に対し, $g_i^p(x_i^*) \geq g_i^p(y_i)$ （式C）

よって, (式C)を示せば良い. 条件(a),(b)を使ってこれを示す.

最適解の条件2：十分性の証明（続き）

関数 $g_i^p(x_i)$ を使って，条件(a), (b) を書き換える：

(a') 任意の $x_i^* > 0$ なる $i \in N$ に対し，$g_i^p(x_i^*) - g_i^p(x_i^* - 1) \geq 0$

(b') 任意の $j \in N$ に対し，$g_j^p(x_j^* + 1) - g_j^p(x_j^*) \leq 0$

したがって，任意の $j \in N$ に対し，

- $x_i^* > 0$ ならば $g_i^p(x_i^* - 1) \leq g_i^p(x_i^*)$ かつ $g_i^p(x_i^*) \geq g_i^p(x_i^* + 1)$.
  よって，凹関数の性質 (i) より，$g_i^p(x_i)$ は $x_i = x_i^*$ において最大.
- $x_i^* = 0$ ならば $g_i^p(0) \geq g_i^p(1)$.
  よって，凹関数の性質 (ii) より，$g_i^p(x_i)$ は $x_i = x_i^*$ において最大.

∴ 任意の $i \in N$ に対し，$g_i^p(x_i^*) \geq g_i^p(y_i)$ ■

命題 （離散）凹関数 $g_i: \mathbb{Z} \to \mathbb{R}$ および整数 α に対し，

(i) $g_i(x_i)$ は $x_i = \alpha$ において最大

⇔ $g_i(\alpha - 1) \leq g_i(\alpha)$ および $g_i(\alpha) \geq g_i(\alpha + 1)$ が成立

(ii) $x_i \geq 0$ の条件下で，$g_i(x_i)$ は $x_i = 0$ において最大

⇔ $g_i(0) \geq g_i(1)$ が成立

関連する問題の最適解との関係

- k=0,1,…,r に対し，次の問題(P_k)を考える
  - 問題(P_r) ＝ 元の資源配分問題そのもの

(P_k) 最大化$\sum_{i=1}^{n} g_i(x_i)$ 条件 $\sum_{i=1}^{n} x_i = k,\ x_i \geq 0$, 整数 $(\forall i = 1, \ldots, n)$

問題(P_k)の最適解から，問題(P_{k+1})の最適解が簡単に計算可能

定理
整数k (0≦k<r)，$(x_1, x_2, \ldots, x_n)$は(P_k)の最適解
j (1≦j≦n) は$g_j(x_j + 1) - g_j(x_j)$の値が最大
➔ x_j を1増やすと，問題(P_{k+1})の最適解になる

➡ 貪欲アルゴリズムの正当性

[証明の概略] $(x_1', \ldots, x_n')$: $(x_1, \ldots, x_n)$ のx_j を1増やしたベクトル
$(x_1^*, \ldots, x_n^*)$: (P_{k+1})の最適解の中で，$\sum_{i=1}^{n} |x_i^* - x_i'|$ 最小
$(x_1^*, \ldots, x_n^*) = (x_1', \ldots, x_n')$ならば証明終わり．
よって$(x_1^*, \ldots, x_n^*) \neq (x_1', \ldots, x_n')$ と仮定，矛盾を導く

様々な制約下での資源配分問題

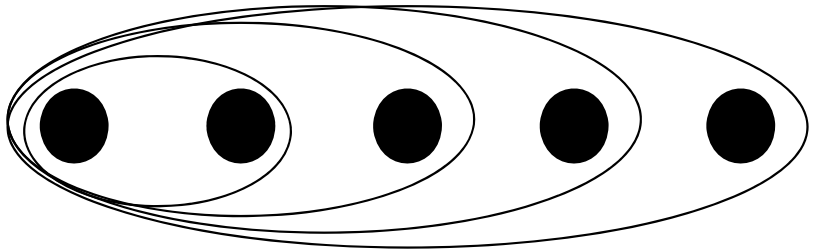
様々な制約

以下の制約を追加しても，貪欲アルゴリズムで最適解が得られる

- 上界制約：各変数の上界$x_i \leq u_i$　（u_i：非負整数）
  - 例：各選挙区の議席数の上限
  - $\sum_{i=1}^{n} u_i \geq r$ ならば実行可能解が存在
- 一般上界制約：{1,2,…,n}を分割した集合$X_1, X_2, \ldots, X_m$に対し，

$$\sum_{i \in X_j} x_i \leq u_j \ (j = 1,2,\ldots,m) \ (u_j：\text{非負整数})$$

  - 例：あるエリアの選挙区全体に割り当てられる議席数の上限
  - $\sum_{j=1}^{n} u_j \geq r$ ならば実行可能解が存在
- 入れ子制約：$\sum_{i=1}^{j} x_i \leq u_j \ (j = 1,2,\ldots,n-1)$ （u_j：非負整数）
  - 例：第1期から第j期の間に消費する労働資源の上限

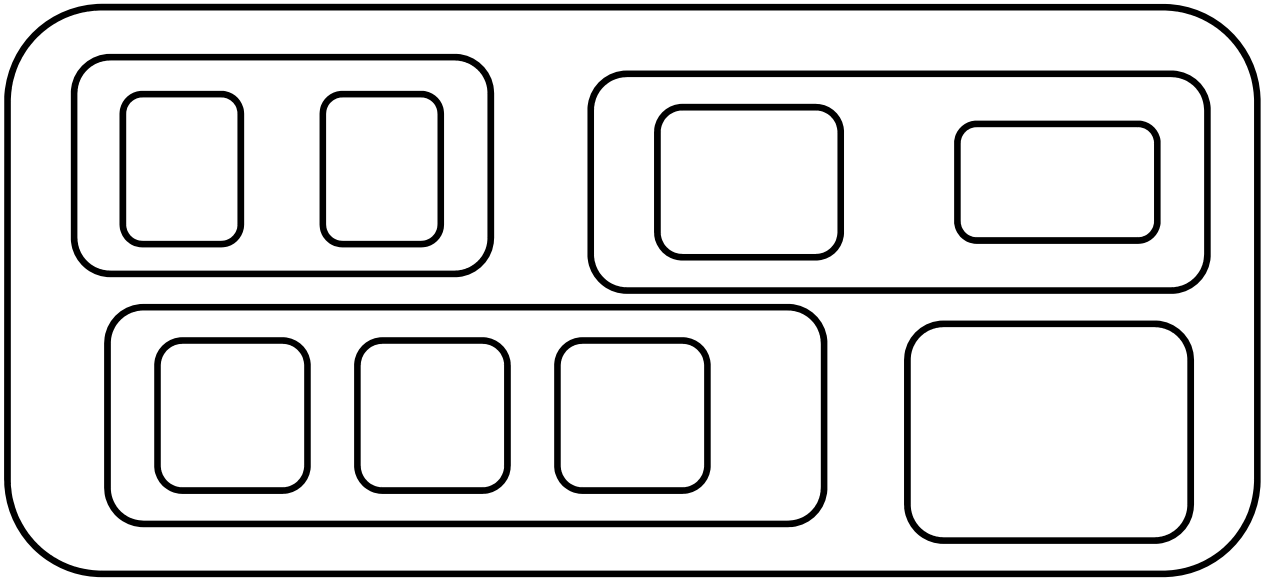


木制約

- 木制約：$X_j \cap X_k = \emptyset$または$X_j \subseteq X_k$または$X_j \supseteq X_k$を満たす

  集合$X_1, X_2, \ldots, X_m \subseteq \{1,2,\ldots,n\}$に対し，

$$\sum_{i \in X_j} x_i \leq u_j \ (j = 1,2,\ldots,m)$$



イメージ

- 各市町村への資源配分量の上限
- 各都道府県の市町村への資源配分量の合計値の上限
- 国内の各地域（東北，関東，など）への資源配分量の合計値の上限

木制約に対する貪欲アルゴリズム

木制約下での資源配分問題に対しても，
貪欲アルゴリズムが適用可能

ステップ0：$(x_1, \ldots, x_n) = (0, \ldots, 0)$とする．
ステップ1：$\sum_{i=1}^{n} x_i = r$ ➔ $(x_1, \ldots, x_n)$ を出力して終了．
ステップ2：$S = \{i | 1 \leq i \leq n, x_i$を1増やしても制約が満たされる$\}$ とおく．
$g_i(x_i + 1) - g_i(x_i)$ が最大の $i \in S$ に対し，x_i を1増やす．
ステップ1に戻る．

正当性の証明は省略

木制約に対する貪欲アルゴリズム：実行例

最大化 $-(2x_1-1)^2-(3x_2-2)^2-3x_3^2$
条件 $x_1+x_2+x_3=6, x_i \geq 0$, 整数
$x_1+x_2 \leq 3, x_1 \leq 4,\ x_2 \leq 2, x_3 \leq 3$

	1	2	3
$g_i(0)$	-1	-4	0
$g_i(1)$	-1	-1	-3
$g_i(2)$	-9	-16	-12
$g_i(3)$	-25	-49	-27
$g_i(4)$	-49	-100	-48

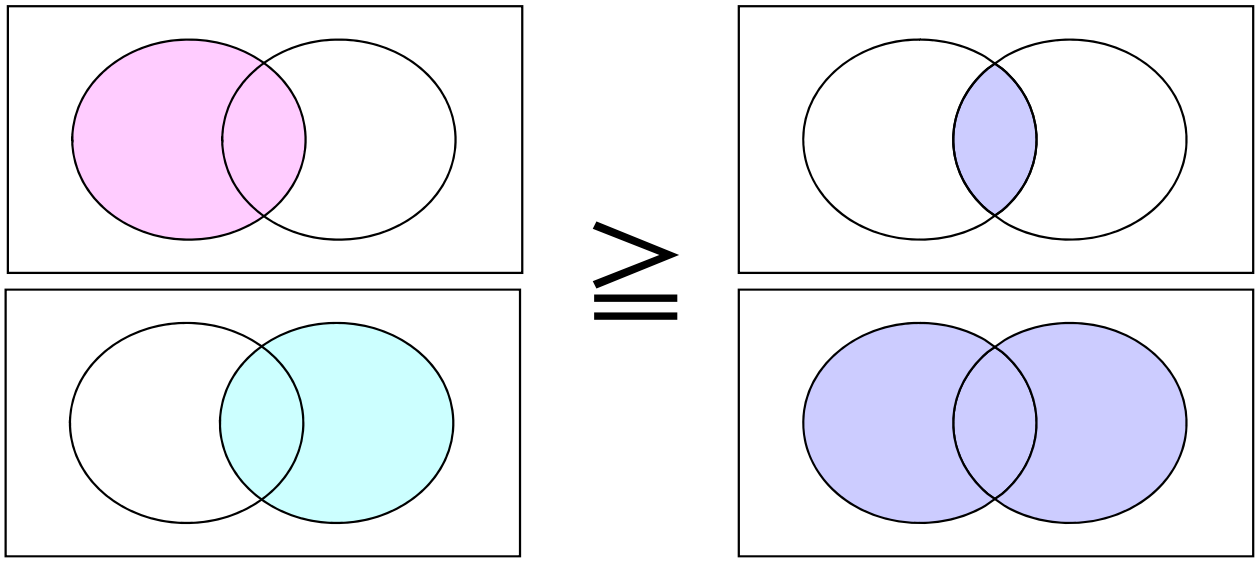
	1	2	3
$g_i(1)-g_i(0)$	0	3	-3
$g_i(2)-g_i(1)$	-8	-15	-9
$g_i(3)-g_i(2)$	-16	-33	-15
$g_i(4)-g_i(3)$	-24	-51	-21

x2→x1→x3→x1→x3→x3
の順に1ずつ増やしていく

劣モジュラ制約

これまで紹介した制約は，劣モジュラ制約の特殊ケース

$N = \{1,2,\dots,n\}$ とおく



劣モジュラ制約：

条件 $\rho(X) + \rho(Y) \geq \rho(X \cap Y) + \rho(X \cup Y)\ \ (\forall X, Y \subseteq N)$ を満たす

$\rho(X) \in \mathbb{Z} \cup \{+\infty\}$ $(X \subseteq \{1,2,\dots,n\})$ に対し，

$$\sum_{i \in X} x_i \leq \rho(X)\ \ (\forall X \subseteq N)$$

制約の具体例：

$x_1 + x_2 \leq 5,\ x_2 + x_3 \leq 3,\ x_2 \leq 1,\ x_1 + x_2 + x_3 \leq 6$