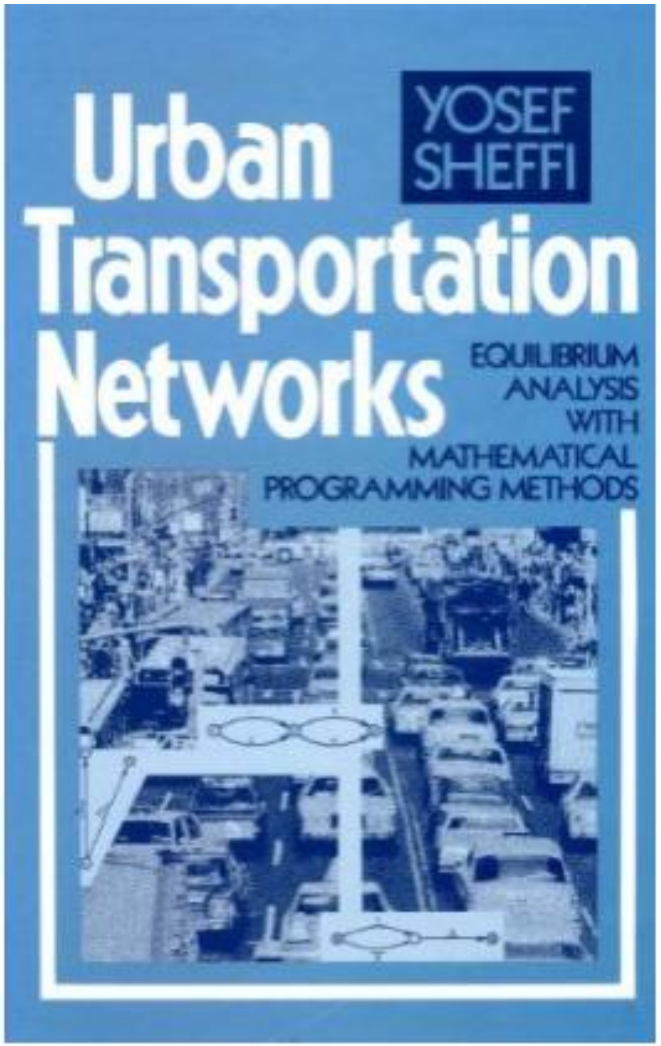
URBAN TRANSPORTATION NETWORKS

Author: Yossi Sheffi

: Prentice Hall, chaputer4～5, October 1985



M1　今泉孝章

Sheffiゼミ

2013/5/16(金)

前回までのおさらい

利用者均衡配分とは?
どのような状況のことを指すか

利用者均衡配分を解くには?
どのように定式化するか

定式化した問題はどうやって解くのか
数学的な最適化の基礎

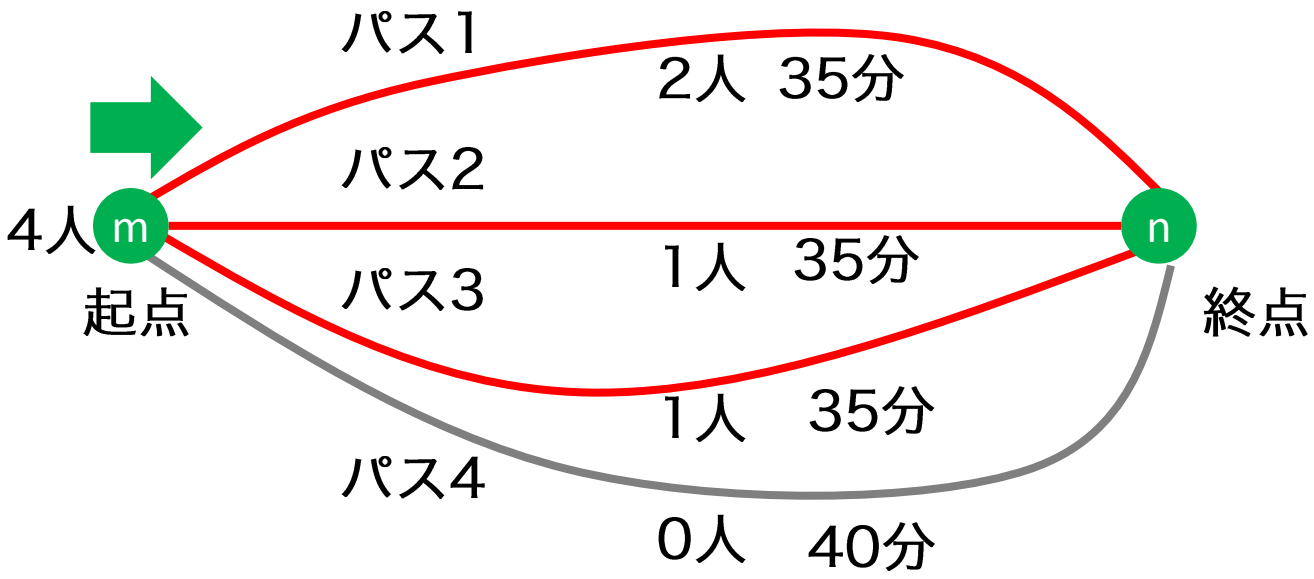
定式化した問題を解くと
利用者均衡が満たされるのか?
最適化手法を使い解いてみる

利用者均衡配分とは?

利用者均衡

全ての旅行者は自分の旅行時間を最小とするように行動する

どの旅行者も経路を変えることによってこれ以上自分の旅行時間を短くすることができない(利用されている経路の旅行時間は同じで、利用されていない経路の旅行時間は利用されている経路の旅行時間より大きいか、せいぜい同じ)



利用者均衡配分を解くには?

変数の定義

a :ネットワーク上のリンク

r :起点ノード

s :終点ノード

k :OD間のパス

f_k^{rs} から導かれるもの

x_a :リンクa上のフロー数

c_k^{rs} :起点 r と終点s を結ぶパスk 上の旅行時間

$\delta_{a,k}^{rs}$:リンクaが起点rと終点sを結ぶパスk に含まれているとき1,そうでないとき0のダミー変数

q_{rs}:起点 r と終点s を結ぶトリップ数

t_a :リンクa 上の旅行時間

与えられるもの

f_k^{rs} :起点 r と終点 s を結ぶパスk上のフロー数

求めたいもの

利用者均衡配分を解くには?

定式化

$$\min\ z(x) = \sum_a \int_0^{x_a} t_a(\omega) d\omega \quad (1.6)$$

$t_a(\omega)$：リンクaにおけるフロー数がωのときの旅行時間

$\sum_a$：ネットワーク上の全リンクの旅行時間の和

$$\sum_k f_k^{rs} = q_{rs} \qquad f_k^{rs} \geq 0 \quad (1.7)$$

$\sum_k f_k^{rs} = q_{rs}$：(1.3)

条件2(フロー数制約)

定式化した問題はどうやって解くのか

$$\min \quad L(x_1, x_2, ..., x_I, u_1, u_2, .., u_J) = z(X) - \sum_{j=1}^{J} u_j \left(\sum_{i=1}^{I} h_{ij} x_i - b_j \right)$$

$$x^* \cdot \frac{\partial L(X^*, U^*)}{\partial x_i} = 0 \qquad \frac{\partial z(X^*, U^*)}{\partial x_i} \geq 0 \qquad \frac{\partial L(X^*, U^*)}{\partial \lambda_j} = 0$$

$$f_k^{rs} \frac{\partial L(f, u)}{\partial f_k^{rs}} = 0 \quad (3.11)$$

$$\frac{\partial L(f, u)}{\partial f_k^{rs}} \geq 0 \quad (3.12)$$

$$\frac{\partial L(f, u)}{\partial u_{rs}} = 0 \quad (3.13)$$

定式化した問題を解くと
利用者均衡が満たされるのか?

$$f_k^{rs}(c_k^{rs} - u_{rs}) = 0$$

$$c_k^{rs} - u_{rs} \geq 0$$

$$\sum_k f_k^{rs} = q_{rs}$$

フロー数制約

$$f_k^{rs} \geq 0$$

最適化問題の解

$$f_k^{rs}(c_k^{rs} - c_{\min}) = 0$$

$$c_k^{rs} - c_{\min} \geq 0$$

$$(c_{\min} : r-s\text{間最短旅行時間})$$

利用者均衡のとき満たされていなければいけない式

以上から均衡配分問題を右のように定式化すれば、その解は利用者均衡配分を満たしていることを証明できた

$$\min \ z(x) = \sum_a \int_0^{x_a} t_a(\omega) d\omega$$

$$\sum_k f_k^{rs} = q_{rs} \quad f_k^{rs} \geq 0$$

制約式

最適化アルゴリズムレビュー

1次元最適化

多変量最適化の解法アルゴリズムの一部となる

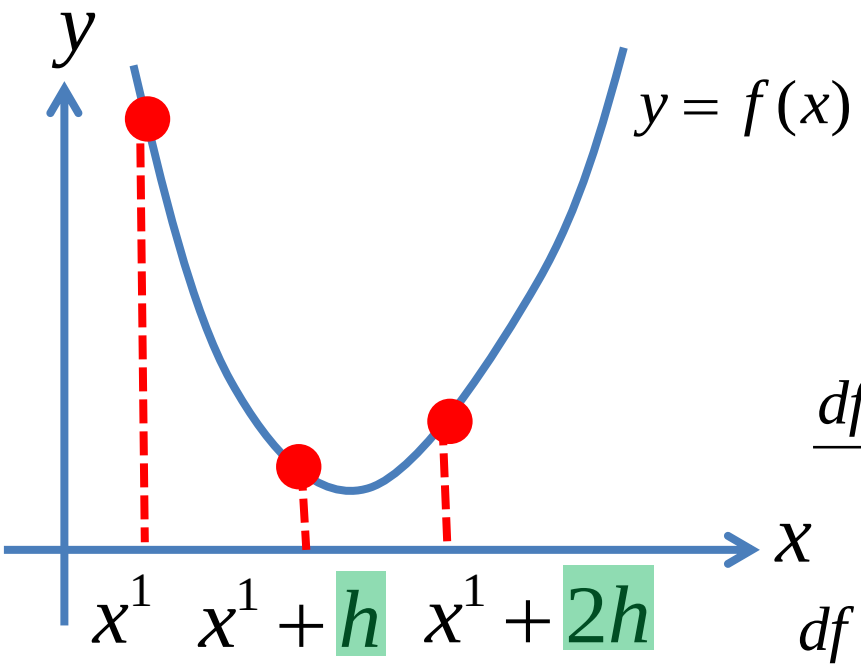
- 区間縮小法
  - 黄金分割法
  - 二等分法
- 曲線当てはめ法
  - ニュートン法
  - 擬点法
  - 2次形式当てはめ法

多変量最適化

降下法を用いたアルゴリズム
最急降下法
凸結合法

最適化アルゴリズムレビュー

最急降下法(制約なし)



$\frac{df(x^1)}{dx} < 0$ ⇒ xを増加させることでもっと関数値は小さくなる

$\frac{df(x^1+h)}{dx} < 0$ ⇒ xを増加させることでもっと関数値は小さくなる

$\frac{df(x^1+2h)}{dx} > 0$ ⇒ xを減少させることでもっと関数値は小さくなる

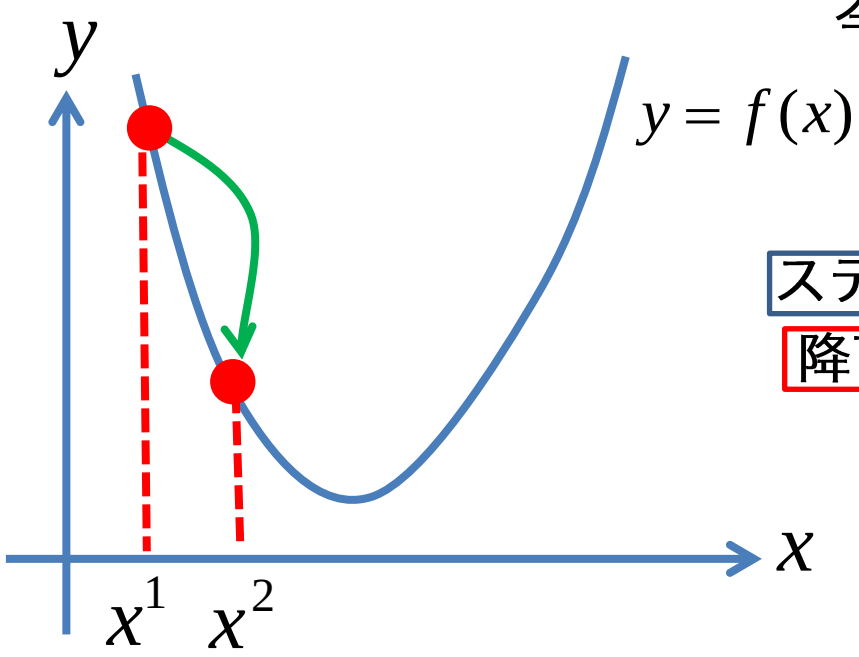
どの方向に動かすべきか?→降下方向

どれくらい動かすべきか?→ステップ幅

降下方向、ステップ幅を変化させて最終的に最小点を見つける

最適化アルゴリズムレビュー

最急降下法(制約なし)



今の位置からどの方向にどれだけ動くか決定

$$x^2 = x^1 + \alpha_1 d^1$$

ステップサイズ　降下方向にどれくらい進むか

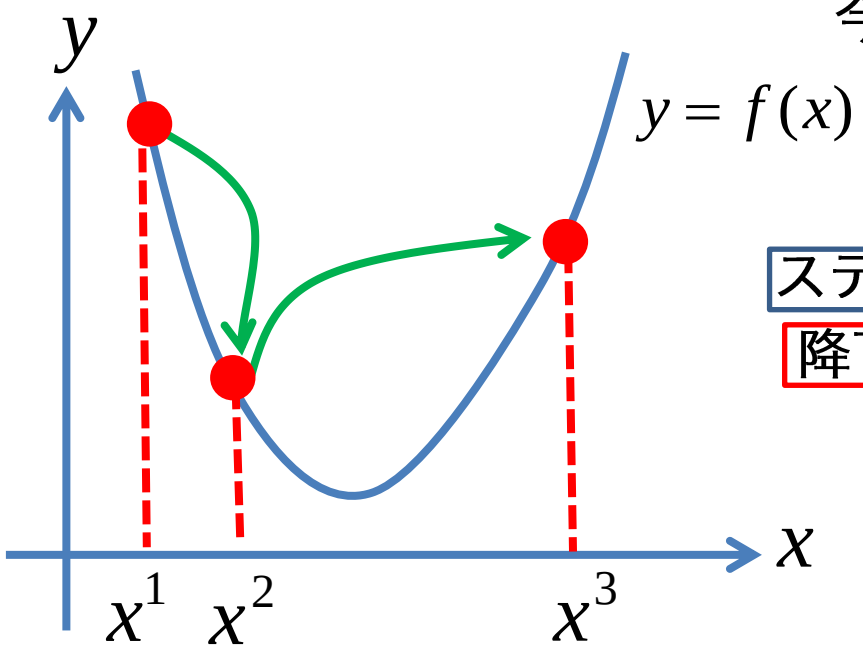
降下ベクトル　目的関数の値が小さくなる方向
(左図の場合xが増加する方向)

$$x^3 = x^2 + \alpha_2 d^2$$

(左図の場合xが増加する方向)

最適化アルゴリズムレビュー

最急降下法(制約なし)



今の位置からどの方向にどれだけ動くか決定

$$x^2 = x^1 + \alpha_1 d^1$$

ステップサイズ　降下方向にどれくらい進むか

降下ベクトル　目的関数の値が小さくなる方向
(左図の場合 x が増加する方向)

$$x^3 = x^2 + \alpha_2 d^2$$

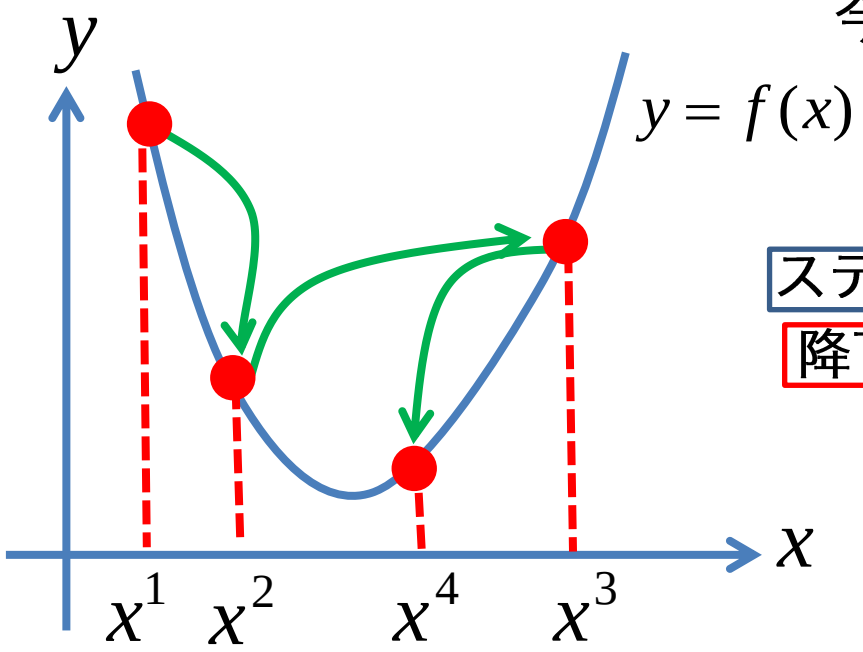
(左図の場合x が増加する方向)

$$x^4 = x^3 + \alpha_3 d^3$$

(左図の場合x が減少する方向)

最適化アルゴリズムレビュー

最急降下法(制約なし)



今の位置からどの方向にどれだけ動くか決定

$$x^2 = x^1 + \alpha_1 d^1$$

ステップサイズ　降下方向にどれくらい進むか

降下ベクトル　目的関数の値が小さくなる方向
(左図の場合 x が増加する方向)

$$x^3 = x^2 + \alpha_2 d^2$$

(左図の場合x が増加する方向)

$$x^4 = x^3 + \alpha_3 d^3$$

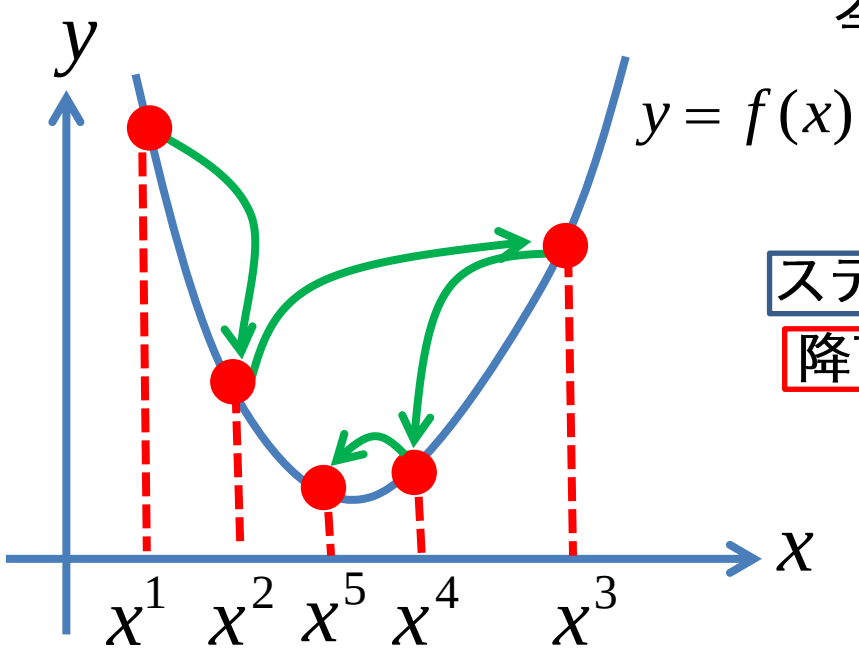
(左図の場合x が減少する方向)

$$x^5 = x^4 + \alpha_4 d^4$$

(左図の場合x が減少する方向)

最適化アルゴリズムレビュー

最急降下法(制約なし)



今の位置からどの方向にどれだけ動くか決定

$$x^2 = x^1 + \alpha_1 d^1$$

ステップサイズ　降下方向にどれくらい進むか

降下ベクトル　目的関数の値が小さくなる方向
(左図の場合xが増加する方向)

$$x^3 = x^2 + \alpha_2 d^2$$

(左図の場合xが増加する方向)

$$x^4 = x^3 + \alpha_3 d^3$$

(左図の場合xが減少する方向)

$$x^5 = x^4 + \alpha_4 d^4$$

(左図の場合xが減少する方向)

この操作をxに関する微分がほとんど0になるまで行う

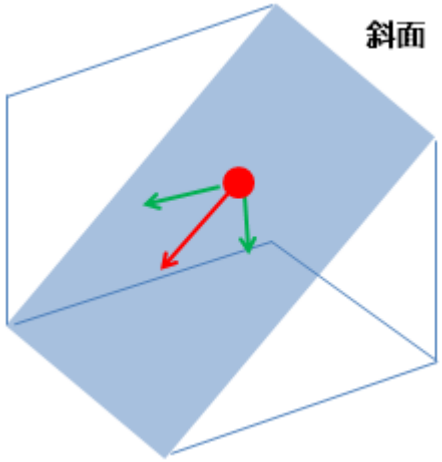
$$\left|\frac{\partial f(x)}{\partial x}\right| \leq \kappa$$

κはあらかじめ決定している微小な定数

ステップサイズと降下ベクトルを効率求めたい

最適化アルゴリズムレビュー

最急降下法(制約なし)



降下方向→最も速く関数の値が小さくなる方向が望ましい
→赤矢印の方向が最も速く下に行ける方向
→勾配ベクトルの逆方向
→目的関数の微分×(-1)

降下方向ベクトル

(一変数) $z(x) = f(x)$

$$d = -\frac{\partial z(x)}{\partial x}$$

(多変数) $z(X) = z(x_1, x_2, .., x_I)$

$$d = -\frac{\partial z(X)}{\partial x_i} = -\nabla z(X)$$

$$i = 1,2,...I$$

n 回目の試行のときの解が $X^n = (x_1^n, x_2^n, ..., x_I^n)$ だとし、降下方向ベクトル、ステップ幅を決定し新たな解 $X^{n+1} = (x_1^{n+1}, x_2^{n+1}, ..., x_I^{n+1})$ をつくる

最適化アルゴリズムレビュー

最急降下法(制約なし)

降下ベクトル:目的関数の値が小さくなる方向
正規化されており、スカラーは1

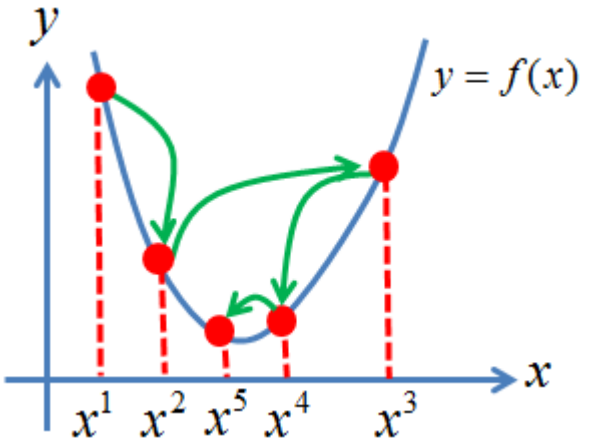
$$X^{n+1} = X^n + \alpha_n d^n$$

勾配ベクトルの反対
=目的関数の微分×(-1)
$= -\nabla z(X)$

ステップサイズ:降下方向にどれくらい進むか

・大きすぎると関数の値が大きくなる可能性がある
(x^2からx^3 に移動した場合)

・小さすぎると関数の値が大きくなる可能性がある
(x^1からx^2 に移動した場合)



最も効率のよい α_n は?

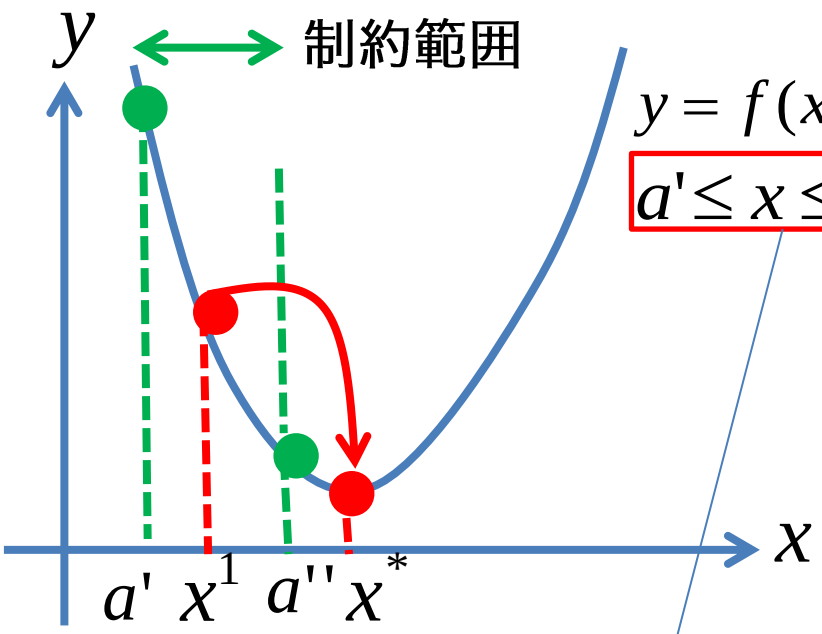
$$\min\ z[X^n + \alpha_n(-\nabla z(X^n))](= z(X^{n+1}))\quad subject\ to\ \alpha_n > 0$$

$$\frac{d}{d\alpha_n} z[X^n + \alpha_n(-\nabla z(X^n))] = 0 \qquad (4.1)$$

凸ならば停留点が最小

最適化アルゴリズムレビュー

最急降下法(制約有り)



制約なしの場合x^1からx^*へ移動したときが最も目的関数の値は小さくなる

⇔制約式があるのでx^*へは移動できない→a''のとこで最小となる

$y = f(x)$

目的関数

$x \ge a'$

$-x \ge -a''$

制約式

→ 移動後に有効でない制約式 $x > a'$

→ 移動後に有効な制約式　$-x = -a''$

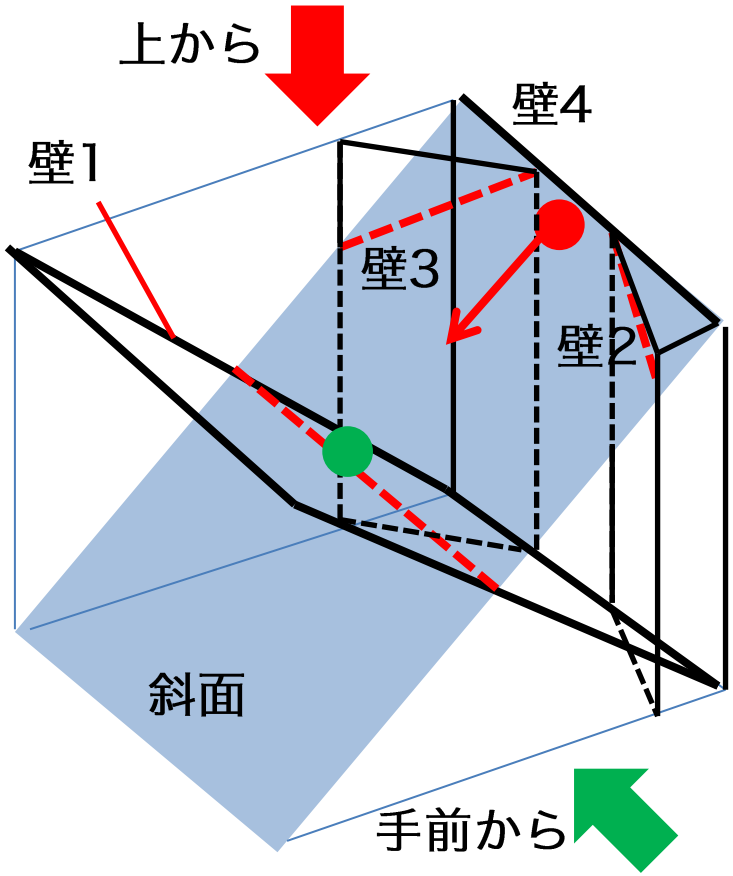
一般化

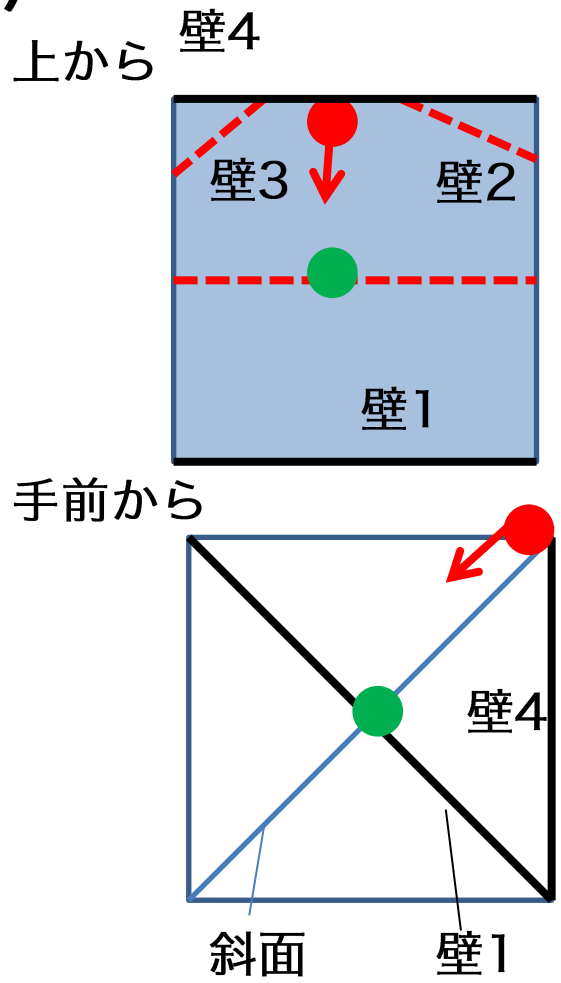
$z(X) = z(x_1, x_2, ..., x_I)$

$\sum_i h_{ij} x_i \ge b_j \quad j = 1,...J \quad (h_{ij}, b_j$は定数$)$

最適化アルゴリズムレビュー

最急降下法(制約有り)

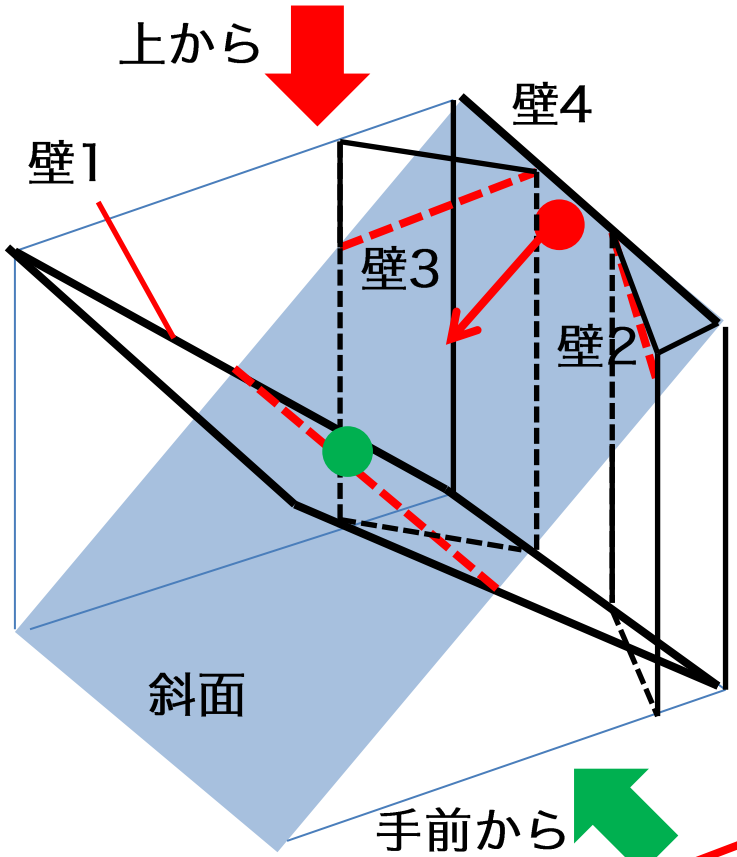


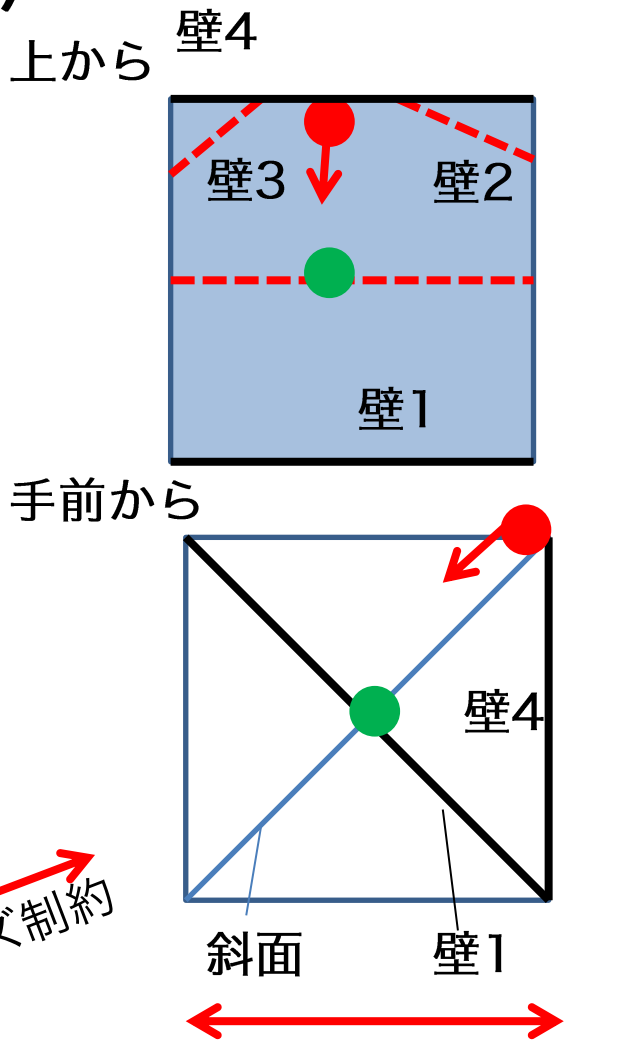


赤矢印が勾配ベクトルの反対方向で最も下へ行ける方向
→どこまで行くか?
(ステップサイズ)

最適化アルゴリズムレビュー

最急降下法(制約有り)



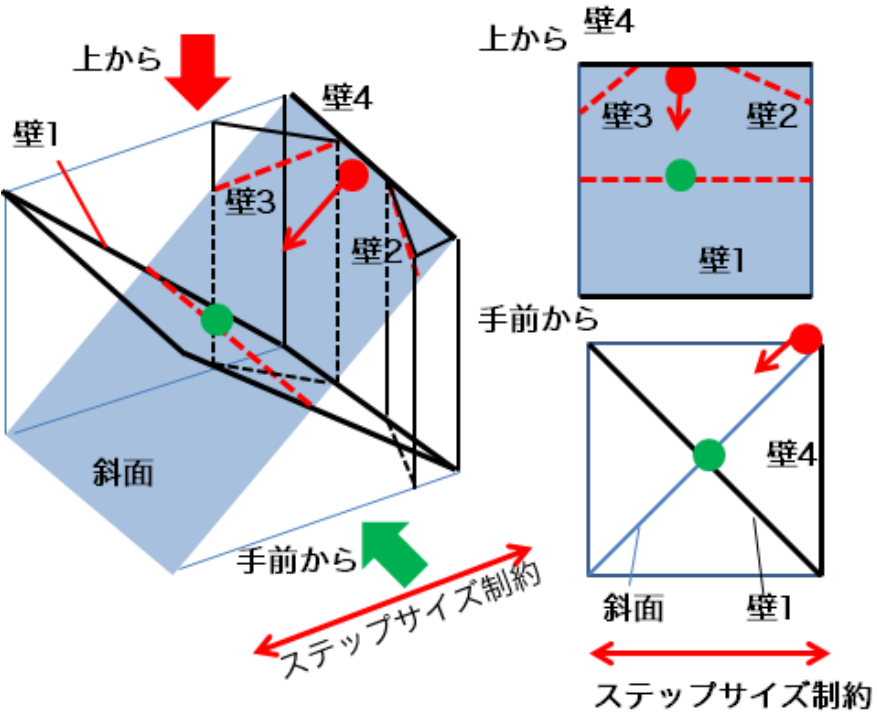


赤矢印が勾配ベクトルの反対方向で最も下へ行ける方向
→どこまで行くか?
(ステップサイズ)
→緑丸まで行く

適当なステップサイズだと制約を超えてしまう

最適化アルゴリズムレビュー

最急降下法(制約有り)



斜面=目的関数

$$z(X) = z(x_1, x_2, \ldots, x_I)$$

ステップサイズ制約

壁=不等式制約

$$\sum_i h_{ij} x_i \geq b_j \quad j = 1, \ldots J \quad (h_{ij}, b_j \text{は定数})$$

赤丸では…

有効な制約式=壁4

破る可能性のある制約式=壁1

破る可能性のない制約式=壁2,3

赤丸= X^n (n回目のときの解)

赤矢印= $-\nabla z(X^n)$

最適化アルゴリズムレビュー

最急降下法(制約有り)

不等式制約(壁)

$$\sum_i h_{ij} x_i^n \geq b_j \quad j=1,\ldots J \quad (h_{ij}, b_j \text{は定数})$$

不等式制約

(ⅰ)有効な制約式(=壁4) $\sum_i h_{ij} x_i^n = b_j \quad \forall j \in J'$ (4.2)　J'を等式を満たす制約式の集合とする

(ⅱ)有効でない制約式(=壁1,2,3) $\sum_i h_{ij} x_i^n > b_j \quad \forall j \in J''$ (4.3)　J''を等式を満たさない制約式の集合とする

$$X^{n+1} = X^n + \alpha_n d^n \Rightarrow \sum_i h_{ij}(x_i^n + \alpha_n d^n) \geq b_j \quad j=1,\ldots J \quad (h_{ij}, b_j \text{は定数})$$

$$\Rightarrow \sum_i h_{ij} x_i^n + \alpha_n \sum_i h_{ij} d_i^n \geq b_j \quad (4.4)$$

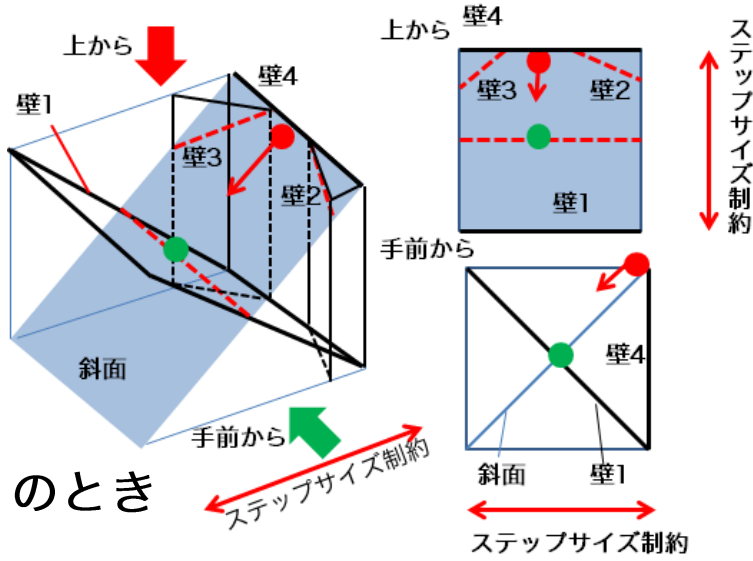
$$\alpha_n > 0$$

どんなステップサイズのときでも(4.2)が満たされなくてはいけない

最適化アルゴリズムレビュー

最急降下法(制約有り)

$$\sum_i h_{ij} x_i^n + \alpha_n \sum_i h_{ij} d_i^n \geq b_j \qquad \alpha_n > 0 \qquad (4.4)$$

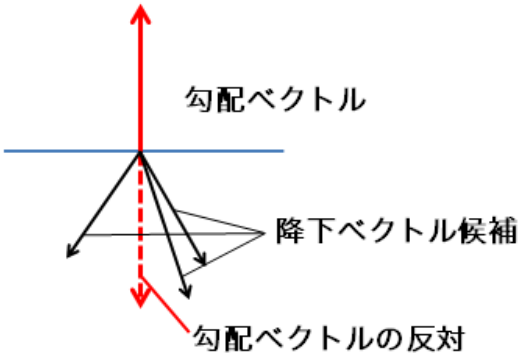


(ｉ)有効な制約式(=壁4) $\sum_i h_{ij} x_i^n = b_j \quad \forall j \in J'$ (4.2)

➡ (4.4)が満たされるのは $\sum_i h_{ij} d_i^n \geq 0$ (4.5) のとき

壁4の制約を破らないための降下方向に関する制約

降下方向は勾配ベクトルの反対向きが最も望ましいが、制約条件によってはその方向に進めない場合もある→(4.5)を満たす方向の中で、勾配ベクトルの反対方向と最も近いものが最適 →勾配ベクトルと成す角度が最大



➡ 勾配ベクトルと降下ベクトルの内積が最小となる

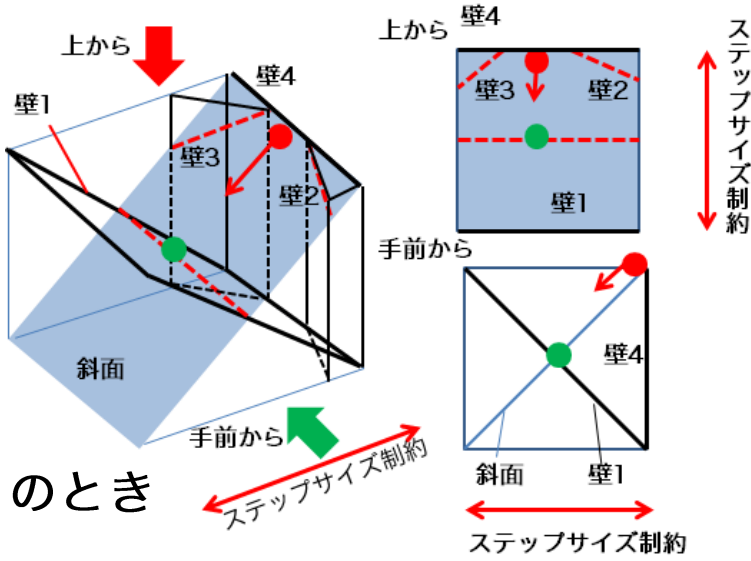
$$\min \nabla z(x^n) \cdot d^T \qquad (4.6)$$

$$subject \quad to \quad \sum_n h_{ij} d_i^n \geq 0 \ \forall j \in J' \ , \sum_i d_i^2 = 1$$

最適化アルゴリズムレビュー

最急降下法(制約有り)

$$\sum_i h_{ij} x_i^n + \alpha_n \sum_i h_{ij} d_i^n \geq b_j \quad (4.4)$$
$$\alpha_n > 0$$



(ⅰ)有効な制約式(=壁4) $\sum_i h_{ij} x_i^n = b_j \quad \forall j \in J'$ (4.2)

➡ (4.4)が満たされるのは $\sum_i h_{ij} d_i^n \geq 0$ (4.5) のとき

壁4の制約を破らないための降下方向に関する制約

(ⅱ)有効でない制約式(=壁1,2,3) $\sum_i h_{ij} x_i^n > b_j \quad \forall j \in J''$ (4.3)

➡ (4.3)より(4.4)は (a) $\sum_i h_{ij} d_i^n > 0$ (4.7) のとき必ず満たされる ➡ 壁2,3

(b) $\sum_i h_{ij} d_i^n < 0$ (4.8) のとき破られる可能性がある ➡ 壁1

(4.3),(4.8)を満たす制約式が最大ステップサイズを決定する

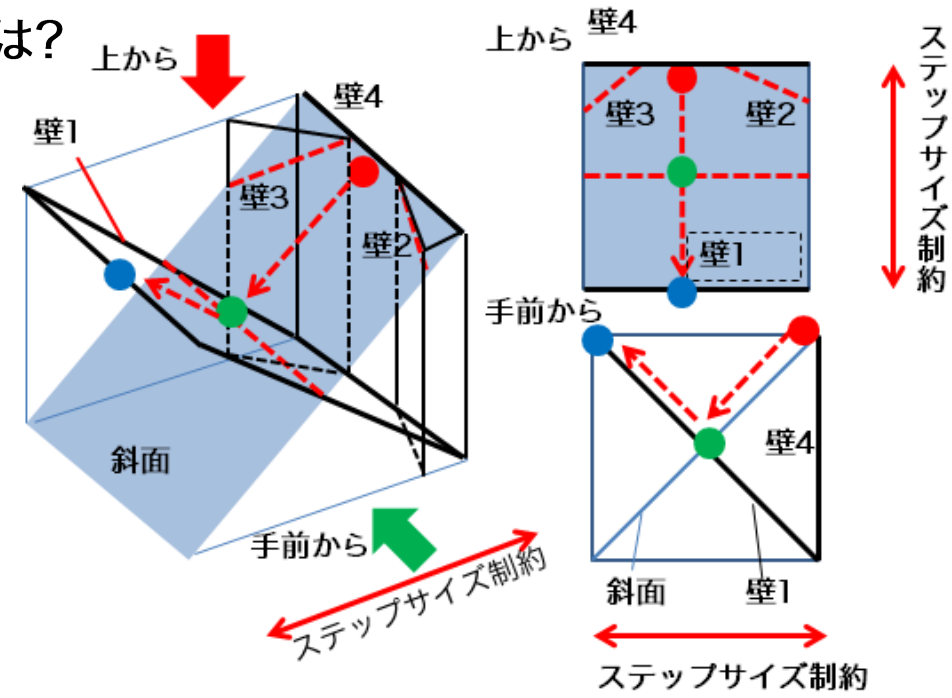
最適化アルゴリズムレビュー

最急降下法(制約有り)

制約式を満たすぎりぎりのステップサイズは?
→移動後に制約式上にある
→壁1の上にある(青丸)

破られる可能性のある制約式((4.3),(4.8)を満たす)の等式が成り立つところが最大ステップ幅

複数破られる可能性のある制約式があるときは、全ての制約式の最大ステップ幅のうち最も小さいものを最大ステップ幅とする(どの制約式も破られない)



制約式の等式が成り立つとき $\sum_i h_{ij} x_i^n + \alpha_n \sum_i h_{ij} d_i^n = b_j$ ⇒ $\alpha_n = \dfrac{b_j - \sum_i h_{ij} x_i^n}{\sum_i h_{ij} d_i^n}$

$$\alpha_n^{\max} = \min_{\forall j \in J'''} \frac{b_j - \sum_i h_{ij} x_i^n}{\sum_i h_{ij} d_i^n} \quad (4.9)$$

(J''' は破られる可能性のある制約式の集合)

最適化アルゴリズムレビュー

最急降下法(制約有り)

まとめ

$$\min \nabla z(x^n) \cdot d^T \qquad (4.6)$$

$$subject \quad to \quad \sum_n h_{ij} d_i^n \geq 0 \ \forall j \in J' \ , \sum_i d_i^2 = 1$$

降下方向

$$\alpha_n^{\max} = \min_{\forall j \in J'''} \frac{b_j - \sum_i h_{ij} x_i^n}{\sum_i h_{ij} d_i^n} \qquad (4.9)$$

ステップサイズ

制約式を吟味して、降下方向、ステップサイズを計算する操作を繰り返すことになる

最適化アルゴリズムレビュー

凸結合法(Frank Wolfe法)

最急降下法・基本的に勾配ベクトルの反対方向に進む→効率が悪い可能性
・ステップサイズを独立に求めるのが煩雑
→簡単にステップサイズを求めかつ効率のよい方向を探索したい

$$\min z(X) = z(x_1, x_2, .., x_I) \qquad \sum_i h_{ij} x_i \geq b_j \quad j = 1,...J \quad (h_{ij}, b_j \text{は定数})$$

n 回目の解が $X^n = (x_1^n, x_2^n, ..., x_I^n)$ のとき $n+1$ 回目の解が $Y^n = (y_1^n, y_2^n, ..., y_I^n)$ だとする。但し、Y^n は許容集合内にあるとする

$$\sum_i h_{ij} y_i \geq b_j \quad j = 1,...J \quad (h_{ij}, b_j \text{は定数}) \qquad (4.10)$$

最適化アルゴリズムレビュー

凸結合法(Frank Wolfe法)

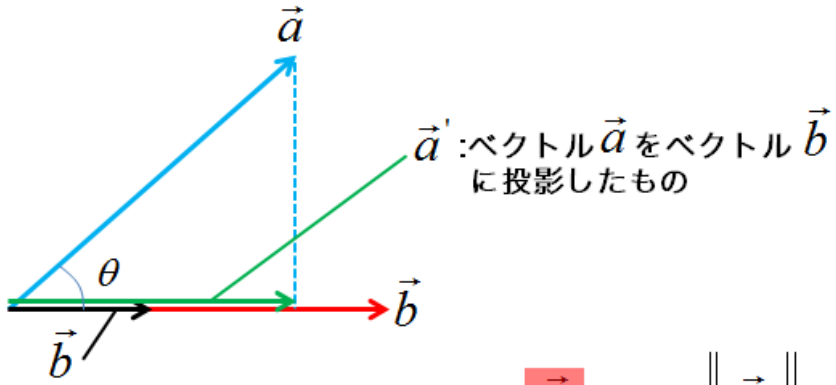
勾配ベクトルの反対方向ベクトル $-\nabla z(x^n)$
を $X^n \rightarrow Y^n$ 方向ベクトル $(Y^n - X^n)$
に投影すると

$$-\nabla z(x^n)\cdot\frac{(Y^n - X^n)^T}{\left\|Y^n - X^n\right\|}$$

この方向に解 $Y^n = (y_1^n, y_2^n, \ldots, y_I^n)$

のあるところ($\left\|Y^n - X^n\right\|$)まで進むと

$-\nabla z(x^n)\cdot(Y^n - X^n)^T$ だけ目的関数は
小さくなる



投影ベクトル=投影したいベクトルと投影する方向の単位ベクトルの内積

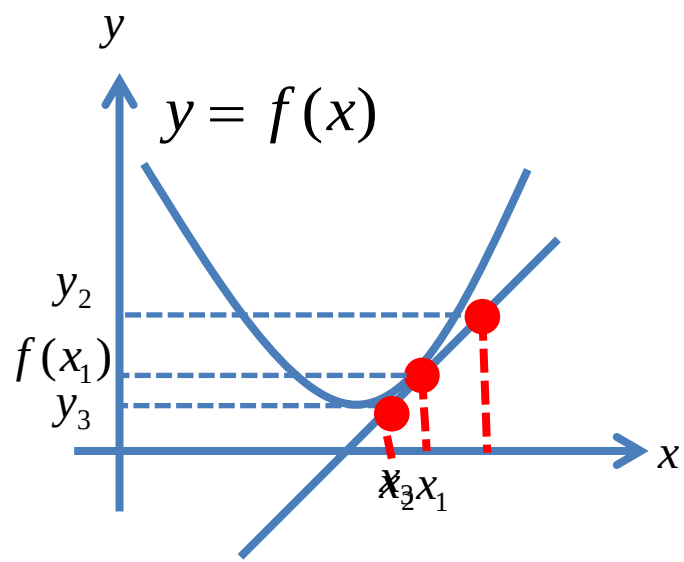
これが最大になればよいので

$$\min \nabla z(X^n)\cdot(Y^n - X^n)^T = \sum_i \frac{\partial z(X^n)}{\partial x_i}(y_i^n - x_i^n) \quad (4.11)$$

$$\sum_i h_{ij} y_i \geq b_j \quad j = 1,\ldots J \quad (h_{ij}, b_j \text{は定数}) \quad (4.10)$$

最適化アルゴリズムレビュー

凸結合法(Frank Wolfe法)



線型近似直線

$y = f'(x_1)(x - x_1) + f(x_1)$

関数の降下方向探索で先ほどは直接関数が小さくなる方向を探した
→線型近似した直線上で探索してもいいのでは?

$$y_2 = f'(x_1)(x_2 - x_1) + f(x_1) > f(x_1)$$

$$y_3 = f'(x_1)(x_3 - x_1) + f(x_1) < f(x_1)$$

目的関数 $z(X) = z(x_1, x_2, .., x_I)$
の線型近似直線を考える
ただし $\sum_i h_{ij} x_i \geq b_j \quad j = 1,...J$ (h_{ij}, b_jは定数)

$X^n = (x_1^n, x_2^n, ..., x_I^n)$における線形近似は

$$z(X^n) + \nabla z(X^n) \cdot (X - X^n)^T$$

未知の点 $Y^n = (y_1^n, y_2^n, ..., y_I^n)$ における線型近似直線の値は

$$z_L^n(Y^n) = z(X^n) + \nabla z(X^n) \cdot (Y^n - X^n)^T$$

最適化アルゴリズムレビュー

凸結合法(Frank Wolfe法)

先ほどと同様に線型近似直線の値が最も小さくなる$Y^n=(y_1^n,y_2^n,...,y_I^n)$を見つければいいので、解くべき問題は

$$\min z_L^n(Y^n)=z(X^n)+\nabla z(X^n)\cdot(Y^n-X^n)^T \quad (4.11)$$

$Y^n=(y_1^n,y_2^n,...,y_I^n)$について定数

$$\sum_i h_{ij}y_i \geq b_j \quad j=1,...J \quad (h_{ij},b_jは定数)$$

制約式

$$\min \nabla z(X^n)\cdot {Y^n}^T=\sum_i\left(\frac{\partial z(X^n)}{\partial x_i}\right)y_i^n \quad (4.12)$$

$$\sum_i h_{ij}y_i \geq b_j \quad j=1,...J \quad (h_{ij},b_jは定数)$$

同様にX^nを消去して

$$\min \nabla z(X^n)\cdot(Y^n-X^n)^T=\sum_i\frac{\partial z(X^n)}{\partial x_i}(y_i^n-x_i^n) \quad (4.11)$$

$$\sum_i h_{ij}y_i \geq b_j \quad j=1,...J \quad (h_{ij},b_jは定数) \quad (4.10)$$

この問題で降下方向を決定できる
→ステップサイズを考える

最適化アルゴリズムレビュー

凸結合法(Frank Wolfe法)

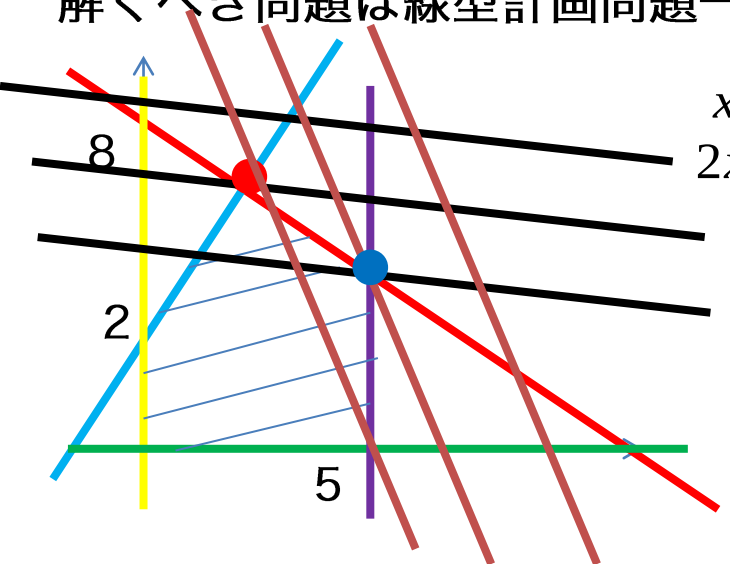
降下する方向は $(Y^n - X^n)$ である
このときステップサイズはどこまでいってもいいのか?

$$\min z_L^n(Y^n) = z(X^n) + \nabla z(X^n)\cdot(Y^n - X^n)^T$$ → 線型関数

$$\sum_i h_{ij} y_i \geq b_j \quad j = 1,...J \quad (h_{ij}, b_j \text{は定数})$$ → 線型関数

制約式

解くべき問題は線型計画問題→その解は必ず制約式の端点となる



$x + y \leq 8$
$2x + 2 \leq y$
$0 \leq x$
$0 \leq y$
$x \leq 5$

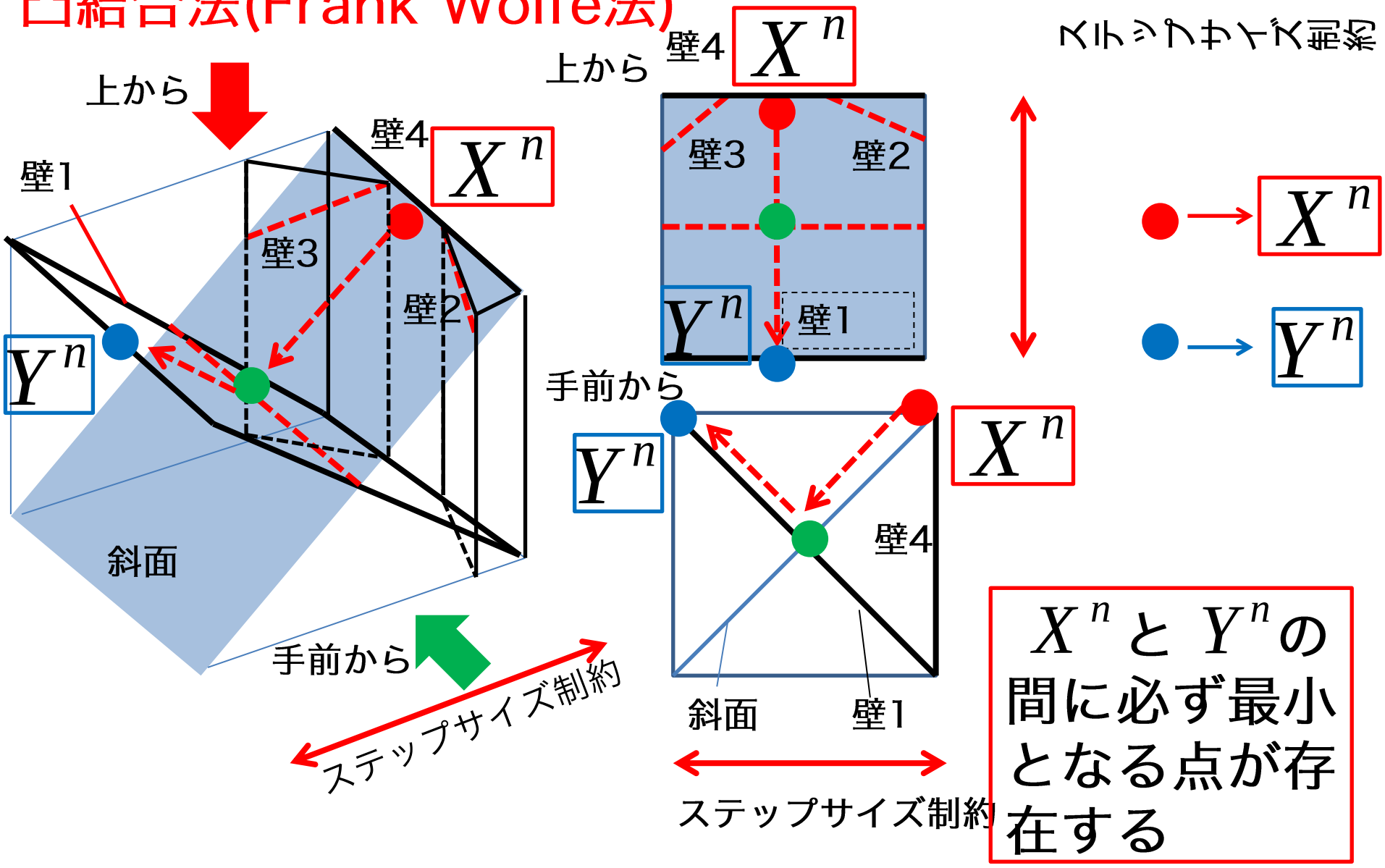
$\max x + 2y$ → 赤丸が解

$\max 3x + y$ → 青丸が解

$Y^n = (y_1^n, y_2^n, ..., y_I^n)$ は制約式上の点となる

最適化アルゴリズムレビュー

凸結合法(Frank Wolfe法)



X^n と Y^n の間に必ず最小となる点が存在する

最適化アルゴリズムレビュー

凸結合法(Frank Wolfe法)

降下方向 $(Y^n - X^n)$ において進むべきステップサイズは

$$\min\ z\left[X^n + \alpha_n\left(Y^n - X^n\right)\right] \quad (4.13)$$

$$0 \le \alpha_n \le 1$$ ← 内分する点なので

↓

値更新

$$X^{n+1} = X^n + \alpha_n\left(Y^n - X^n\right)$$ 探索方向、ステップサイズをもとに値を更新

収束判定

$z(X^n) - z(X^{n+1}) \le \kappa$ → 解を決定し、試行を終了

$z(X^n) - z(X^{n+1}) > \kappa$ → $n+2$ 回目の試行を行う

最適化アルゴリズムレビュー

凸結合法(Frank Wolfe法)

まとめ

$$\min\ z(X) = z(x_1, x_2, \ldots, x_I)$$

$$\sum_i h_{ij} x_i \geq b_j \quad j = 1, \ldots J \quad (h_{ij}, b_j \text{は定数})$$

1. 適当な初期値$X^1 = (x_1^1, x_2^1, \ldots, x_I^1)$ を与え

$$\min \nabla z(X^n) \cdot Y^{nT} = \sum_i \left(\frac{\partial z(X^n)}{\partial x_i} \right) y_i^n \quad (4.12)$$

$$\sum_i h_{ij} y_i \geq b_j \quad j = 1, \ldots J \quad (h_{ij}, b_j \text{は定数})$$

により $Y^1 = (y_1^1, y_2^1, \ldots, y_I^1)$を求める

2.ステップサイズを決定する

$$\min z\left[X^n + \alpha_n \left(Y^n - X^n\right)\right] \quad (4.13)$$

$$0 \leq \alpha_n \leq 1$$

3.収束判定により試行終了か継続かを決定

$$z(X^n) - z(X^{n+1}) \leq \kappa$$

利用者均衡を解く

$$\min\ z(X)=\sum_a \int_0^{x_a} t_a(\omega)d\omega \qquad \sum_k f_k^{rs}=q_{rs} \quad f_k^{rs}\geq 0$$

制約式

$$X=(x_1,x_2,\ldots,x_I)$$

関数の微分

$$\min z^n(Y)=\nabla z(X^n)\cdot Y^T=\sum_a\left(\frac{\partial z(X^n)}{\partial x_a}\right)y_a$$

補助変数

$$Y=(y_1,y_2,\ldots,y_I)$$

y_a は x_a の補助変数とみなせる

$$\frac{\partial z(X^n)}{\partial x_a}=t_a^n=t_a(x_a^n) \qquad x_a=\sum_r\sum_s\sum_k f_k^{rs}\delta_{a,k}^{rs}$$ なので

なので目的関数 $z(X)$ は f_k^{rs} の関数ともみなせる

→ f_k^{rs} の補助変数を g_k^{rs} とすると

関数の微分

$$\min z^n(G)=\nabla_f z(x(F^n))\cdot G^T \qquad \sum_{rs}\sum_k g_k^{rs}=q_{rs} \quad g_k^{rs}\geq 0$$

補助変数

$$F=(f_1^{r_1s_1},f_2^{r_1s_1},\ldots,f_l^{r_ns_m}) \quad G=(g_1^{r_1s_1},g_2^{r_1s_1},\ldots,g_l^{r_ns_m})$$

利用者均衡を解く

$$\min z^n(G) = \nabla_f z(x(F^n)) \cdot G^T = \sum_{rs} \sum_{k} c_k^{rsn} g_k^{rs} \qquad \sum_{rs} \sum_{k} g_k^{rs} = q_{rs} \quad g_k^{rs} \geq 0$$

制約式 (5.2)

$$\frac{\partial}{\partial f_l^{mn}} z[x(f)] = \sum_b \frac{\partial z(x)}{\partial x_b} \frac{\partial x_b}{\partial f_l^{mn}}$$

b:リンクを表す(aと同様)

$$\frac{\partial z(x)}{\partial x_b} = \frac{\partial}{\partial x_b} \sum_a \int_0^{x_a} t_a(\omega) d\omega = t_b \qquad \frac{\partial x_b}{\partial f_l^{mn}} = \delta_{b,l}^{mn} \quad \because (3.1)$$

$$\frac{\partial}{\partial f_l^{mn}} z[x(f)] = \sum_b t_b \delta_{b,l}^{mn} = c_l^{mn}$$

←パス間旅行時間は全リンクの旅行時間の総和

$$\frac{\partial x_a(f)}{\partial f_l^{mn}} = \frac{\partial}{\partial f_l^{mn}} \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs} = \delta_{a,l}^{mn} \quad (3.1)$$

$\sum_k g_k^{rs} \delta_{a,k}^{rs}$ からリンクフローに関

ステップサイズが $\min_{0 \leq \alpha_n \leq 1} z\left[X^n + \alpha_n (Y^n - X^n)\right]$ で決定

$X^{n+1} = X^n + \alpha_n (Y^n - X^n)$ で値を更新

$\dfrac{\sqrt{\sum_a (x_a^{n+1} - x_a^n)^2}}{\sum_a x_a^n} \leq \kappa$ (κは定数) で収束判定