Urban Transportation Network (Yossi Sheffi, 1985) Chap.3 & Chap.4 (p.56~110)

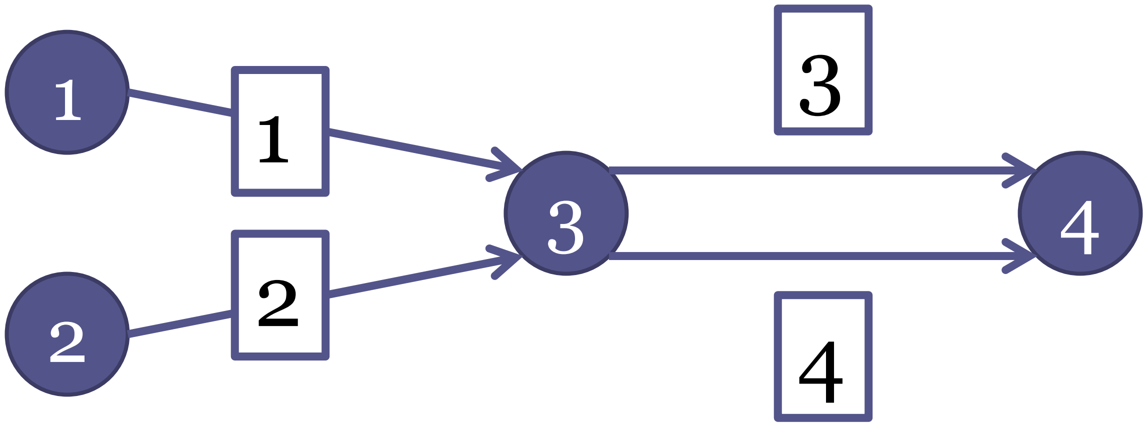
交通研究室　学部４年
日下部　達哉

Contents (Chap.3)

1. 交通網の表現方法
2. User Equilibrium
3. System Optimization
4. 両者の違いは？
5. Braess’s Paradox

1. 交通網の表現方法

<u>今回の例</u>



①から④へ　リンク１→リンク３（ルート１）
　　　　　　リンク１→リンク４（ルート２）
②から④へ　リンク２→リンク３（ルート３）
　　　　　　リンク２→リンク４（ルート４）

1. 交通網の表現方法

<u>変数の定義</u>

q_{rs} …始点Rと終点Sの間の交通流量

x_a …リンクAの交通流量

$t_a(x_a)$ …リンクAの所要時間

f_k^{rs} …始点Rと終点Sを結ぶルートKの交通流量

c_k^{rs} …始点Rと終点Sを結ぶルートKの所要時間

1. 交通網の表現方法

<u>変数の定義（つづき）</u>

$\delta_{a,k}^{rs}$ …始点Rと終点Sを結ぶルートKに

リンクAが含まれる → 1 , 含まれない → 0

Δ …第(a,k)成分に$\delta_{a,k}^{rs}$を入れた行列

1. 交通網の表現方法

<u>リンク⇔ルート間の「交通流量」の関係</u>

…あるリンクを通る全てのルートの交通流量の和が、あるリンクの交通流量。

$$x_a = \sum_r \sum_s \sum_k (f_k^{rs} * \delta_{a,k}^{rs})$$

$\boldsymbol{x} = \boldsymbol{f} * \Delta^{\boldsymbol{T}}$ … ベクトル表記

1. 交通網の表現方法

<u>リンク⇔ルート間の「移動時間」の関係</u>

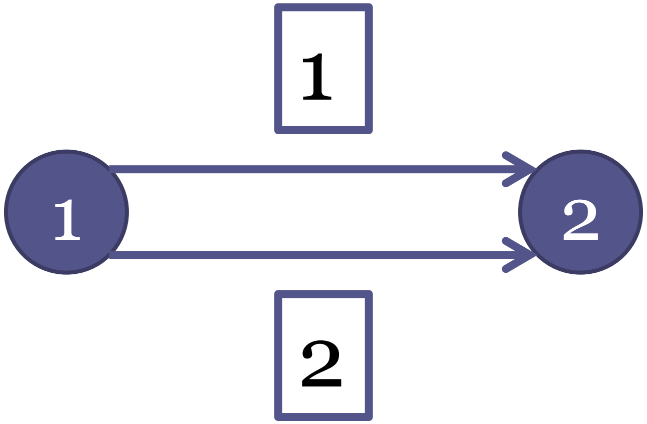
…あるルートが通る全てのリンクの移動時間の和が、あるルートの移動時間。

$$c_k^{rs} = \sum_a (t_a * \delta_{a,k}^{rs})$$

$\boldsymbol{c} = \boldsymbol{t} * \Delta$ …ベクトル表記

2. User Equilibrium

今回の例



$$t_1(x_1) = 10x_1 + 10 \quad (\text{旧道})$$

$$t_2(x_2) = x_2 + 100 \quad (\text{バイパス})$$

$$x_1 + x_2 = 100$$

2. User Equilibrium

そのまま解くと…

$$① \ t_1 = 10x_1 + 10$$

$$② \ t_2 = x_2 + 100$$

$$③ \ x_1 + x_2 = 100$$

$$④ \ t_1 = t_2$$

$$x_1 = \frac{190}{11}, x_2 = \frac{910}{11}, t_1 = t_2 = \frac{2010}{11}$$

2. User Equilibrium

定式化すると…

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

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

個々人の移動時間を最小化

2. User Equilibrium

なぜ？

$$f_k^{rs} * \frac{\partial L(\boldsymbol{f}, \boldsymbol{u})}{\partial f_k^{rs}} = 0 \ \rightarrow\rightarrow \ f_k^{rs} * (c_k^{rs} - u_{rs}) = 0$$

$$\frac{\partial L(\boldsymbol{f}, \boldsymbol{u})}{\partial f_k^{rs}} \geq 0 \ \rightarrow\rightarrow \ c_k^{rs} - u_{rs} \geq 0$$

$$\frac{\partial L(\boldsymbol{f}, \boldsymbol{u})}{\partial u_{rs}} = 0 \ \rightarrow\rightarrow \ \sum_k f_k^{rs} = q_{rs}$$

2. User Equilibrium

これを解いてみると…

$$z(\boldsymbol{x}) = \int_0^{x_1} (10\omega + 10)\, d\omega + \int_0^{100-x_1} (\omega + 100)\, d\omega$$

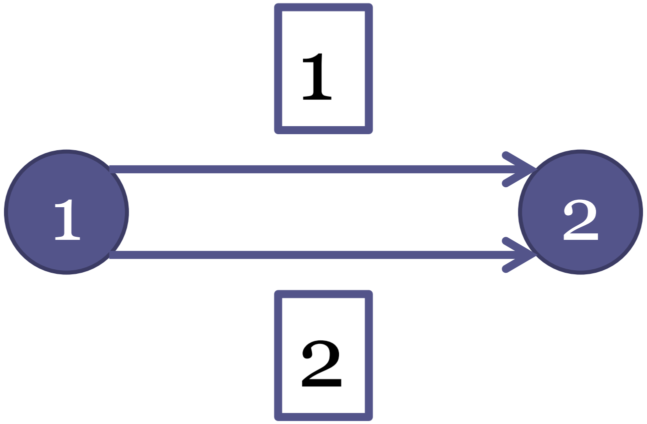
$$x_1 * (11x_1 - 190) = 0$$

$$11x_1 - 190 \geq 0$$

$$x_1 = \frac{190}{11}, x_2 = \frac{910}{11}, t_1 = t_2 = \frac{2010}{11}$$

3. System Optimization

<u>今回の例</u>



$$t_1(x_1) = 10x_1 + 10 \quad (\text{旧道})$$

$$t_2(x_2) = x_2 + 100 \quad (\text{バイパス})$$

$$x_1 + x_2 = 100$$

3. System Optimization

<u>定式化すると…</u>

$$min.\ \ z(\boldsymbol{x}) = \sum_{a} (x_a * t_a(x_a))$$

$$subject\ to\ \ \sum_{k} f_k^{rs} = q_{rs}\ , \quad f_k^{rs} \geq 0$$

全体の総移動時間の合計を最小化

3. System Optimization

これを解いてみると…

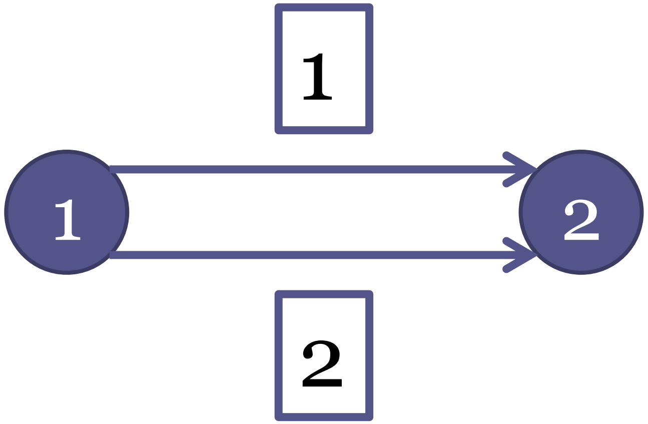
$$z(\boldsymbol{x}) = x_1 * (10x_1 + 10) + (100 - x_1) * (200 - x_1)$$

$$x_1 * (11x_1 - 145) = 0$$

$$11x_1 - 145 \geq 0$$

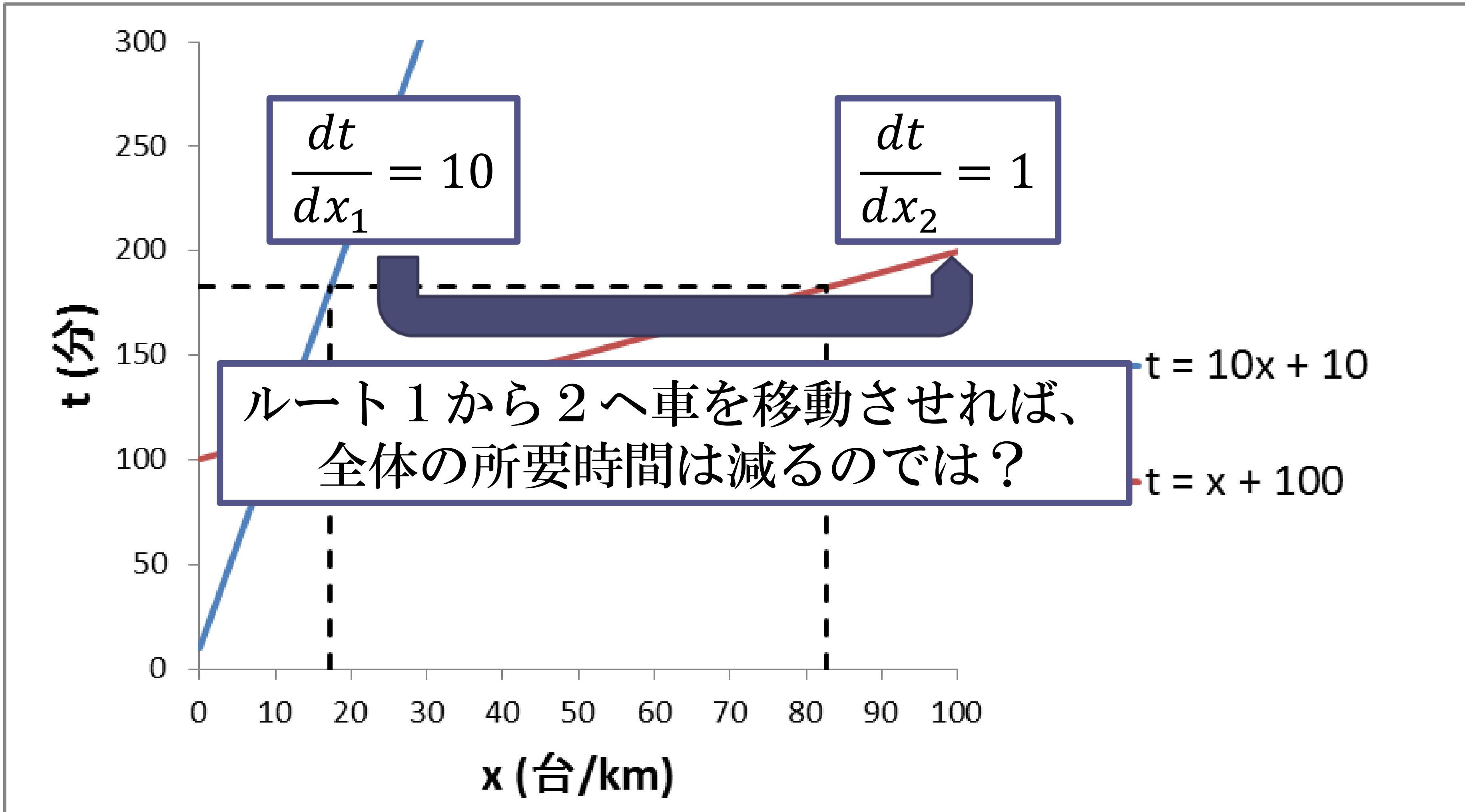
$$x_1 = \frac{145}{11}, x_2 = \frac{955}{11}, t_1 = \frac{1560}{11}, t_2 = \frac{2055}{11}$$

4. 両者の違いは？

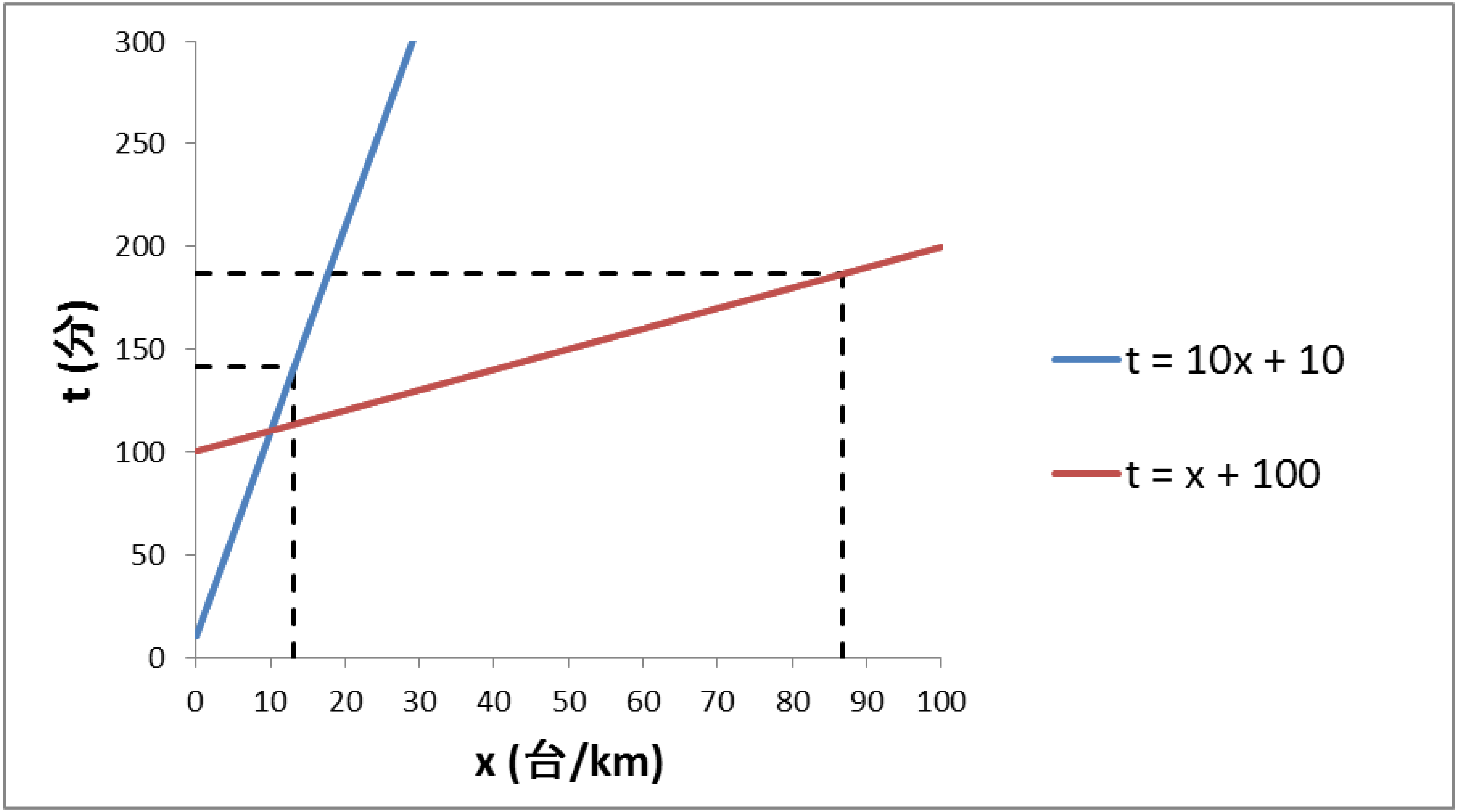


- UE…「個々人が」移動時間を最小化
  →交通網全体として最適かというと…
- SO…「全体で」移動時間を最小化
  →個々人にとって最適かというと…

User Equilibrium



System Optimization



System Optimization

ルート１で減少する総移動時間 … $x_1 * \frac{dt}{dx_1}$

ルート２で増加する総移動時間 … $x_2 * \frac{dt}{dx_2}$

道を変えた人が被る移動時間の増加 … $t_2(x_2) - t_1(x_1)$

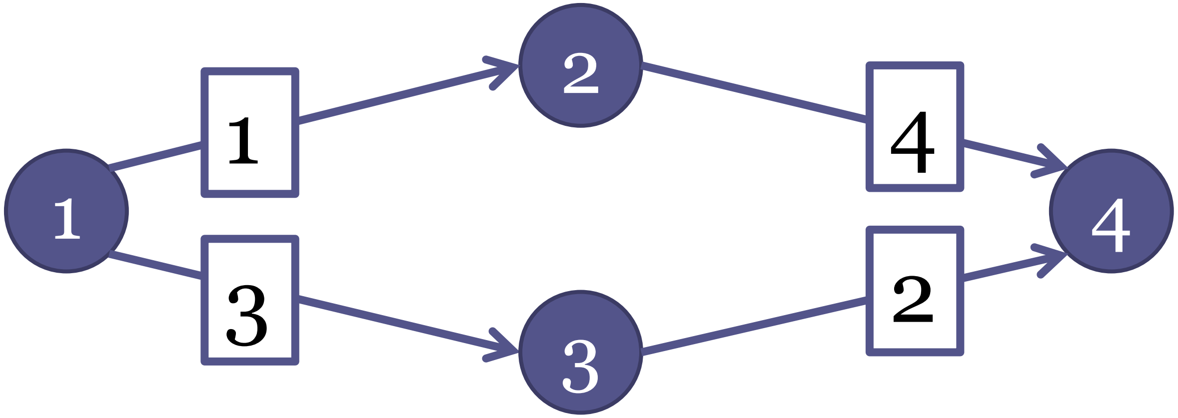
$$\mathrm{t}_1(x_1) + x_1 * \frac{dt}{dx_1} = t_2(x_2) + x_2 * \frac{dt}{dx_2}$$

5. Braess's Paradox

- 道路ネットワークを拡張すると混雑がひどくなる、というパラドックス
- 最短時間で目的地に着こうとする
→新しくできた道へ人が「集まりすぎる」
- 道の利用を制限することで、全体の混雑が逆に緩和される、という可能性がある

5. Braess’s Paradox

これを…



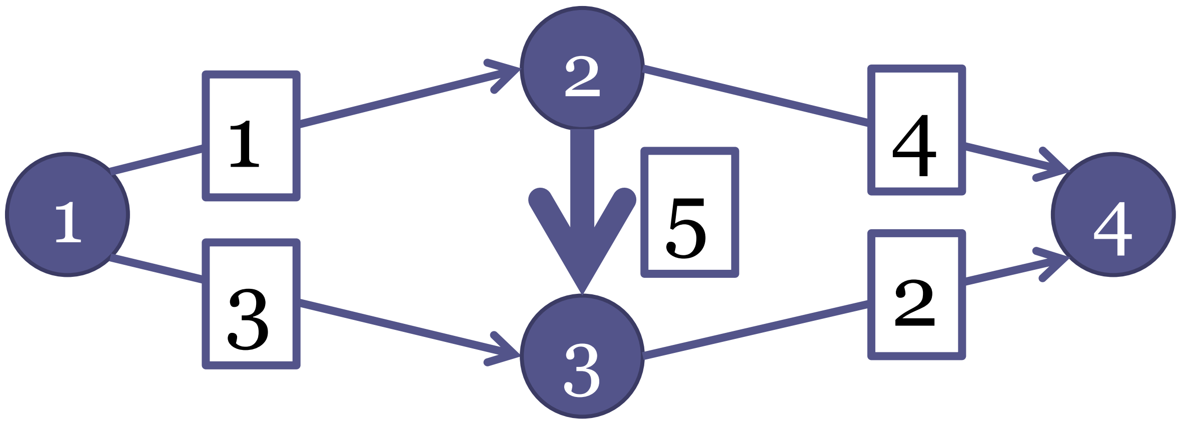
$q_{14} = 6$

$t_1(x_1) = x_1 + 50$ $\mathrm{t}_2(x_2) = x_2 + 50$

$t_3(x_3) = 10x_3$ $t_4(x_4) = 10x_4$

5. Braess's Paradox

こうする。



$q_{14} = 6$

$t_1(x_1) = x_1 + 50$ $\quad$ $t_2(x_2) = x_2 + 50$

$t_3(x_3) = 10x_3$ $\quad$ $t_4(x_4) = 10x_4$

$t_5(x_5) = x_5 + 10$

5. Braess's Paradox

道を作る前

$$min.\ \ z(\boldsymbol{x}) = \sum_{a=1}^{4} \int_{0}^{x_a} t_a(\omega) d\omega$$

$$subject\ to\ \ \sum_{k=1}^{2} f_k = 6\,, \quad f_k \geq 0$$

$$(c_1, c_2, f_1, f_2) = (83{,}83{,}3{,}3)$$

$$T_{total} = 498$$

5. Braess's Paradox

道を作った後

$$min.\ \ z(\boldsymbol{x}) = \sum_{a=1}^{4} \int_{0}^{x_a} t_a(\omega)d\omega$$

$$subject\ to\ \ \sum_{k=1}^{3} f_k = 6\,, \quad f_k \geq 0$$

$$(c_1, c_2, c_3, f_1, f_2, f_3) = (92,92,92,2,2,2)$$

$$T_{total} = 552$$

Contents (Chap.4)

6. 一変数の最適化アルゴリズム

<u>Interval Reduction Methods</u>

- 一定の決まりに従って、最小値を含みえない範囲を切り捨てていく
- 最小値を含みうる範囲が十分に狭くなった時点で、計算を終了する
- 関数が ditonic (単調減少／増加が一回だけ切り替わる) である必要あり

6. 一変数の最適化アルゴリズム

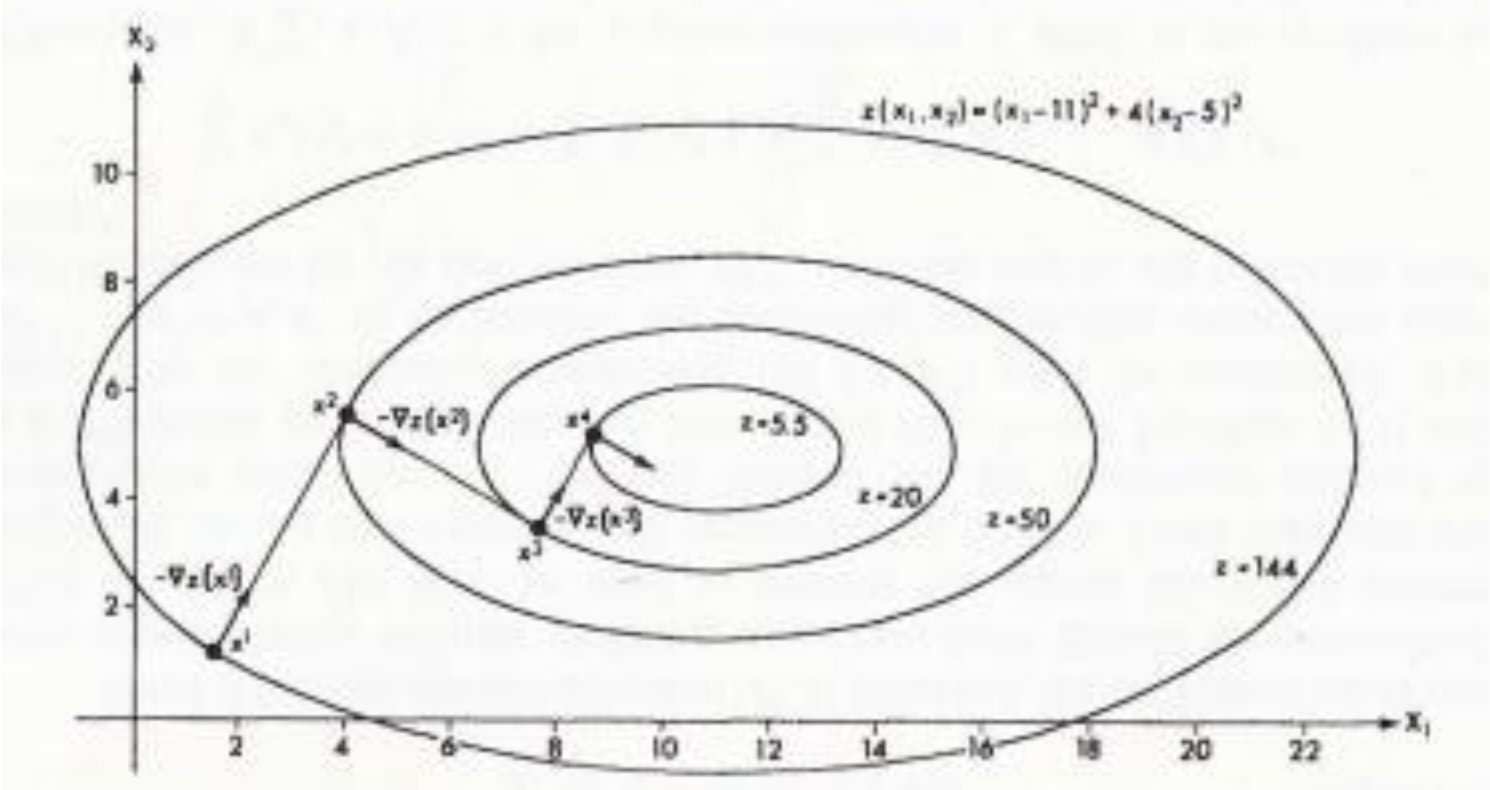
Curve-fitting Methods

- ある点において、対象となる関数に対して
 近似的な二次関数を求める
- その近似的な二次関数を最小にする点を、
 次の点とする
- 点が動く範囲が十分に狭くなった時点で、
 計算を終了する
- 関数が ditonic かつ smooth である必要あり

7. 多変数の最適化アルゴリズム

The Method of Steepest Descent

- ある地点で最も急勾配な方向へ移動を始める
- その方向での最小値に到着したら、
  その地点で最も急勾配な方向へ移動を始める
- 点が動く範囲が十分に狭くなった、または
  勾配ベクトルが十分に0に近づいた時点で、
  計算を終了する



①方向を決める　$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n \nabla z(\boldsymbol{x}_n)$

②位置を決める　$\dfrac{d}{d\alpha} z(\boldsymbol{x}_{n+1}) = 0$

8. 制約下での最適化アルゴリズム

問題

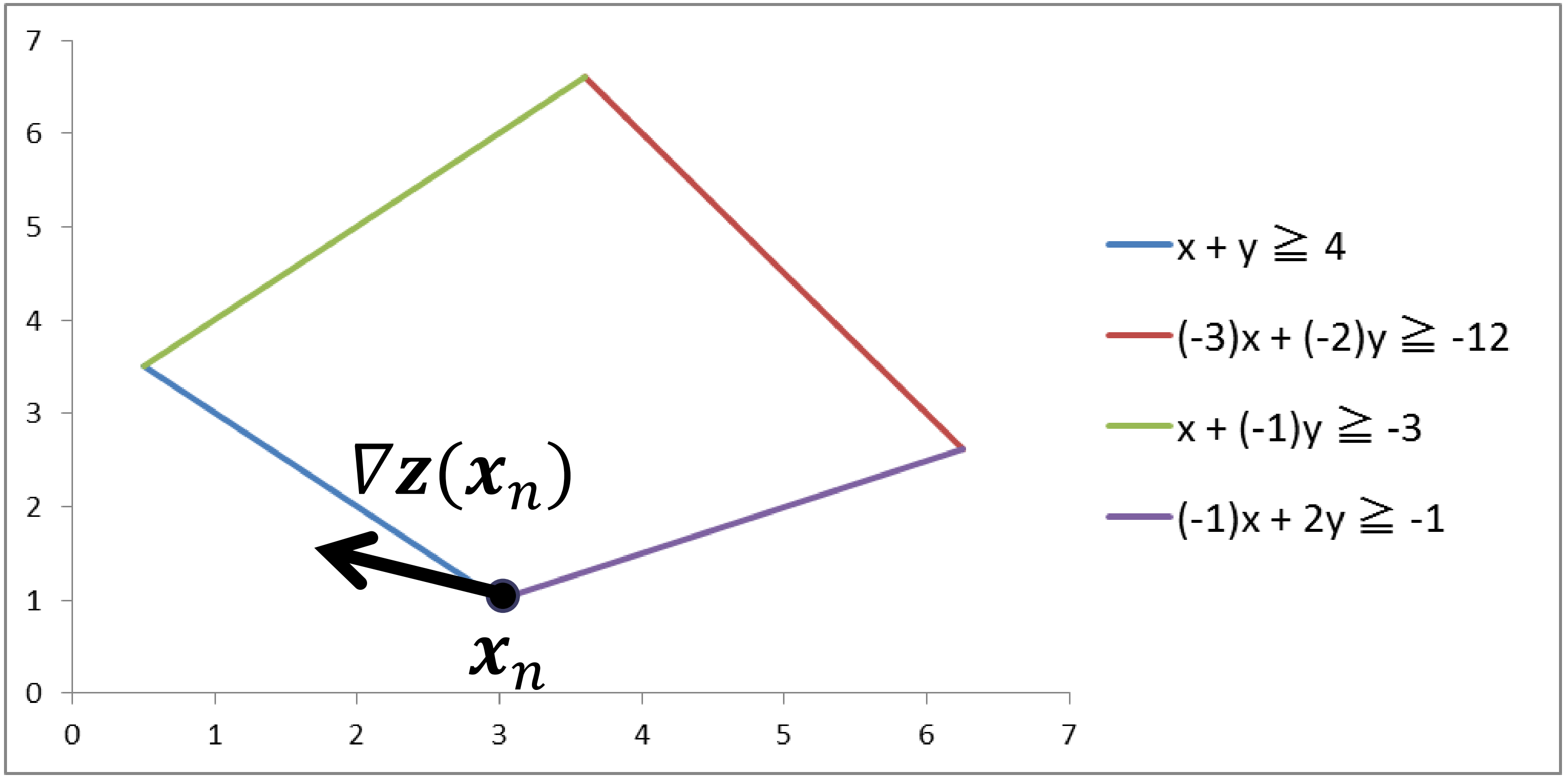
$$min.\ \ z(\boldsymbol{x})$$

$$subject\ to\ \ \boldsymbol{h}_j * \boldsymbol{x} \geq b_j$$

（$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n * \boldsymbol{d}_n$ と表すこととする）

8. 制約下での最適化アルゴリズム

①ベストな $\boldsymbol{d}_n$ の方向を探す



8. 制約下での最適化アルゴリズム

①ベストな $\boldsymbol{d}_n$ の方向を探す

$\boldsymbol{h}_j * \boldsymbol{x}_n = b_j$ …$\boldsymbol{x}_n$が制約式 j の境界線上にある

$\boldsymbol{h}_j * \boldsymbol{x}_n > b_j$ …$\boldsymbol{x}_n$が制約式 j の境界線上にない

$\boldsymbol{d}_n$ を探す際に重要なのは「前者」

… $\boldsymbol{x}_n$ をほんの少し動かした時に、

破られるおそれのある制約は「前者」

8. 制約下での最適化アルゴリズム

①ベストな $\boldsymbol{d}_n$ の方向を探す

$$min.\ \ \nabla \boldsymbol{z}(\boldsymbol{x}_n) * \boldsymbol{d}_n$$

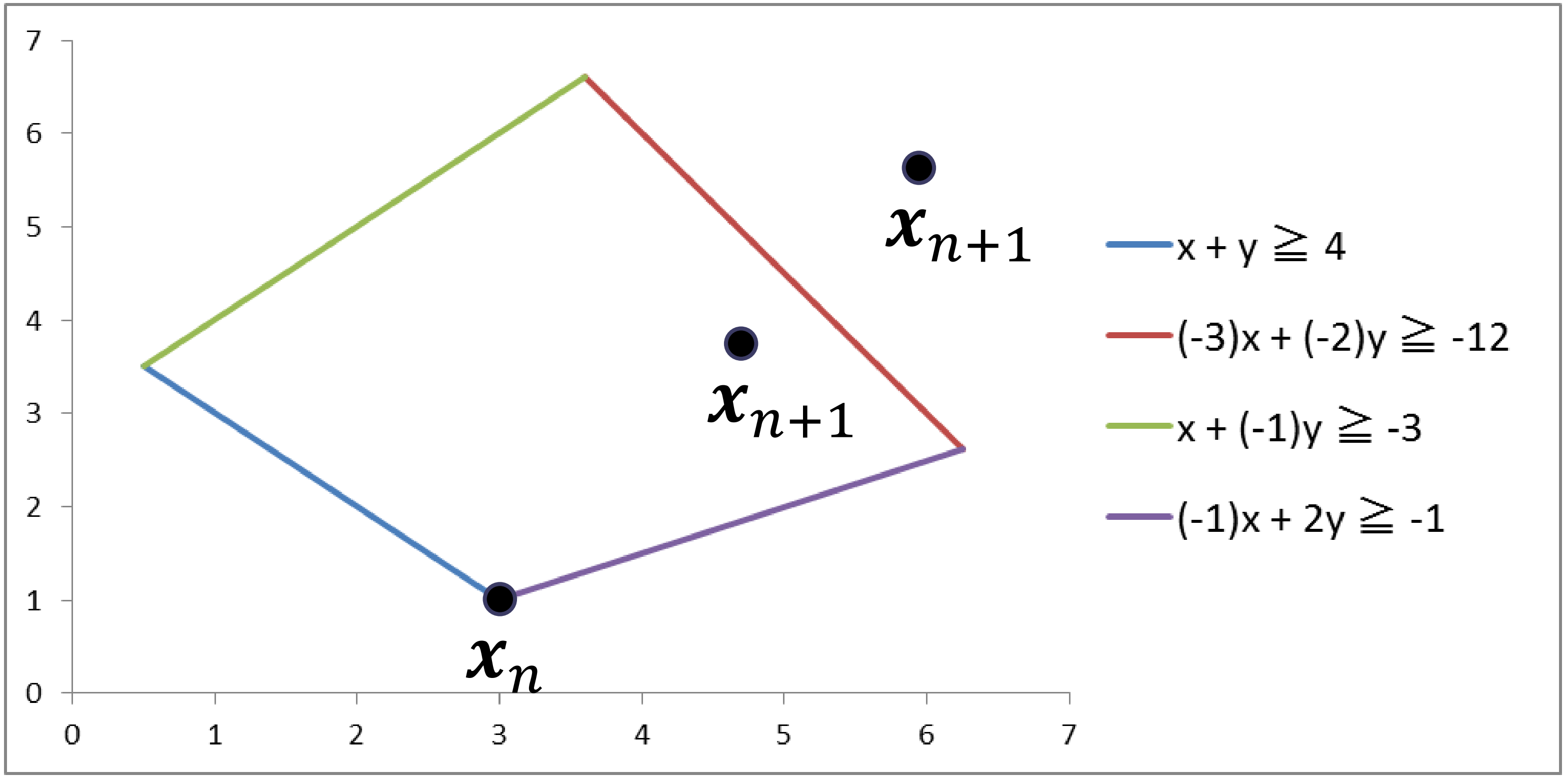
$$subject\ to\ \ \boldsymbol{h}_j * \boldsymbol{x}_n \geq b_j\ ,\quad |\boldsymbol{d}_n| = 1$$

これは $\mathbf{h}_j * \mathbf{x}_n = \mathrm{b}_j$ を満たす j に関してのみ

どっちに行くかが決まった！

8. 制約下での最適化アルゴリズム

②守るべき α_n の範囲を探す



8. 制約下での最適化アルゴリズム

②守るべき α_n の範囲を探す

$\boldsymbol{h}_j * \boldsymbol{x}_n = b_j$ …$\boldsymbol{x}_n$が制約式 j の境界線上にある

$\boldsymbol{h}_j * \boldsymbol{x}_n > b_j$ …$\boldsymbol{x}_n$が制約式 j の境界線上にない

$\boldsymbol{\alpha}_n$ を探す際に重要なのは「前者」

…「後者」の制約を満たさない場合、

そもそも $\boldsymbol{d}_n$ の設定が間違っている

8. 制約下での最適化アルゴリズム

②守るべきα_nの範囲を探す

$$\boldsymbol{h}_j * (\boldsymbol{x}_n + \alpha_n * \boldsymbol{d}_n) \geq b_j$$

$$\alpha_n^{max} = min. \frac{b_j - \boldsymbol{h}_j * \boldsymbol{x}_n}{\boldsymbol{h}_j * \boldsymbol{d}_n}$$

これは $\mathbf{h}_\mathrm{j} * \mathbf{x} > \mathrm{b}_\mathrm{j}$ を満たすjに関してのみ

8. 制約下での最適化アルゴリズム

②守るべきα_nの範囲を探す

$$min.\ \ \boldsymbol{z}(\boldsymbol{x}_n + \alpha_n + \boldsymbol{d}_n)$$

$$subject\ to\ \ 0 \le \alpha_n \le {\alpha_n}^{max.}$$

どこまで向かうかが決まった！