数理計画法　第11回

ネットワーク最適化
非線形計画

担当：　塩浦昭義
(情報科学研究科　准教授)
shioura@dais.is.tohoku.ac.jp

復習:最小費用フロー問題

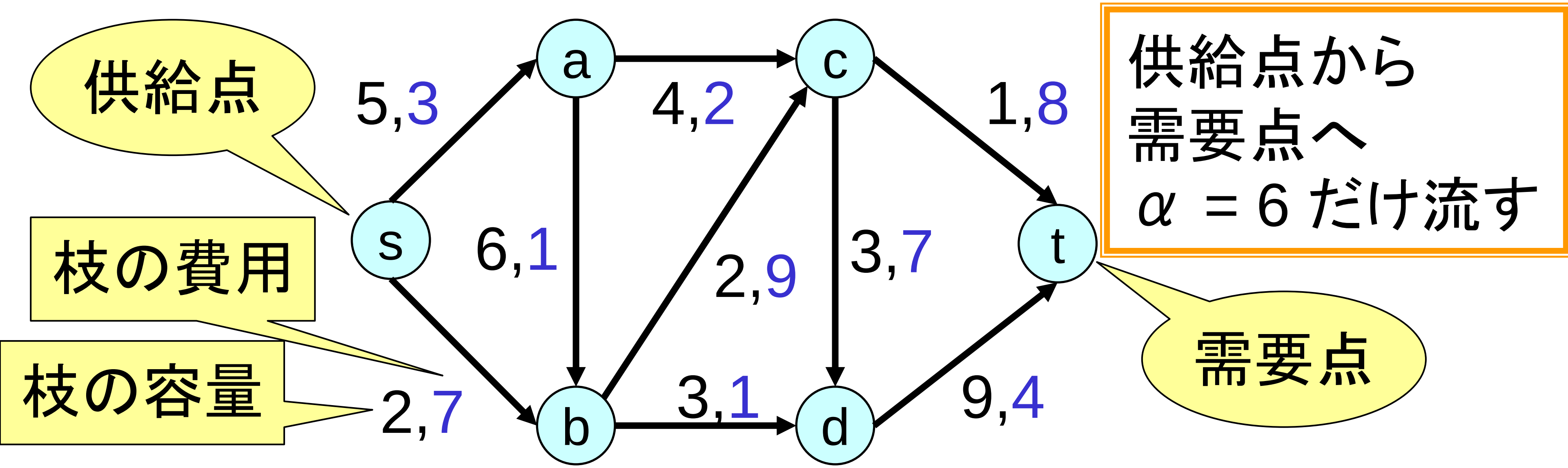
入力:有向グラフ G = (V, E)

供給点 $s \in V$, 需要点 $t \in V$,

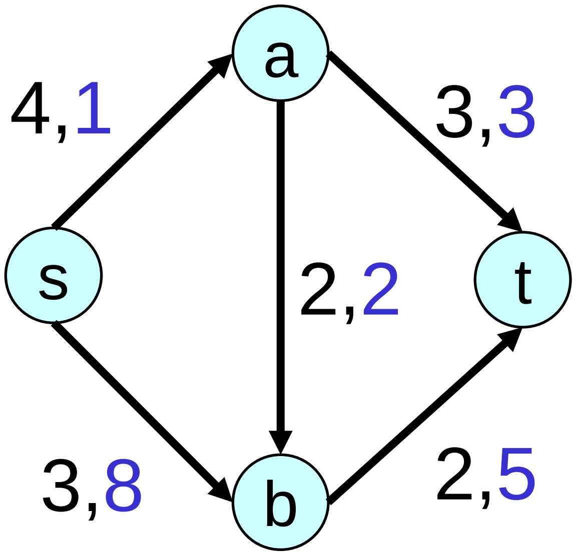
需要(供給)量 $\alpha > 0$

各枝 $(i, j) \in V$ の容量 $u_{ij} \geqq 0$, 費用 c_{ij}

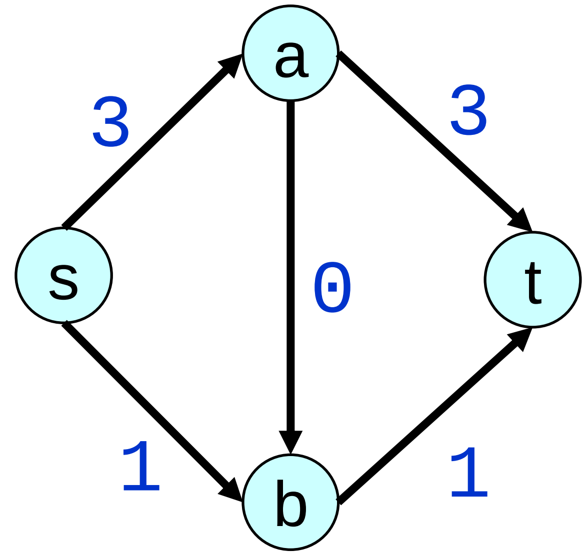
出力:需要供給を満たすフローで総費用が最小のもの



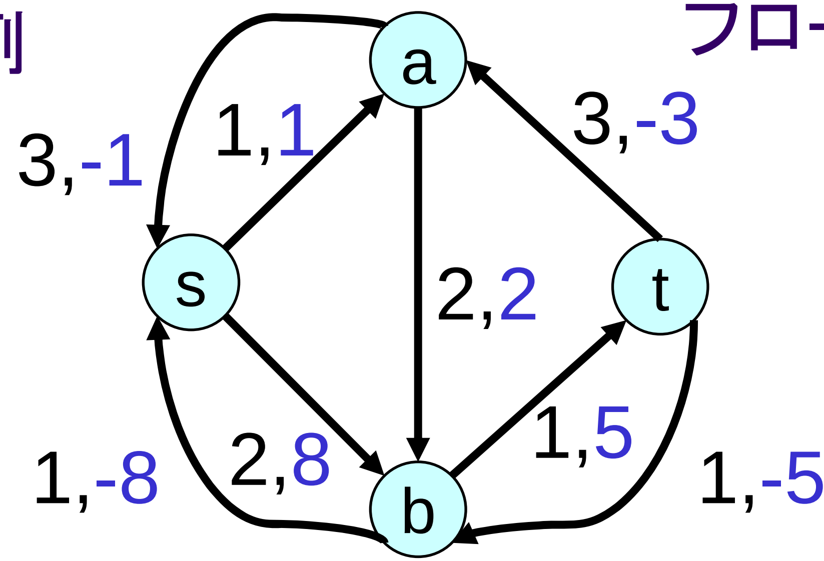
復習:残余ネットワークの作り方



問題例



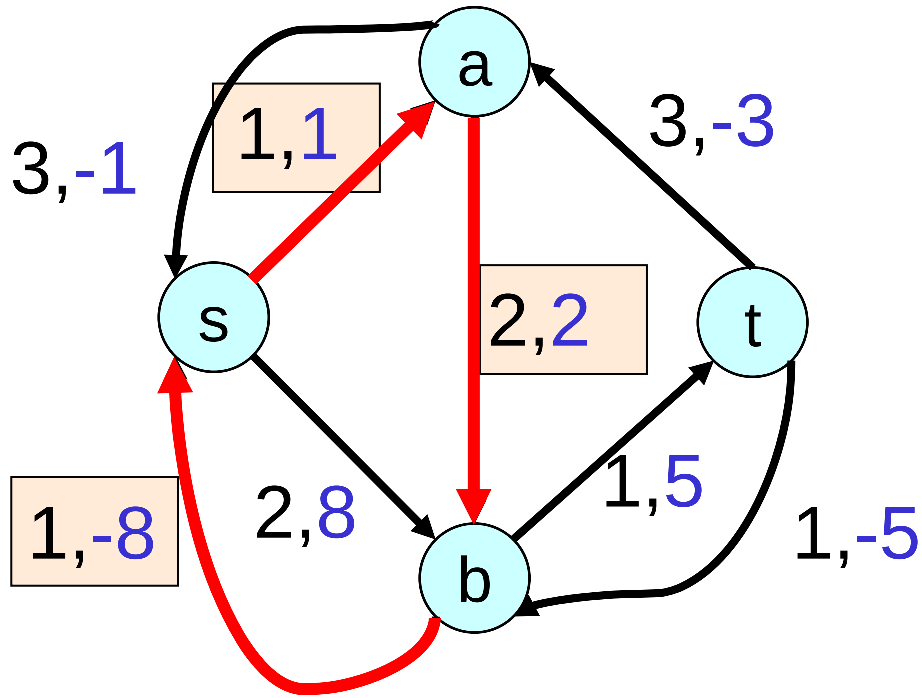
フローの例



残余ネットワーク

復習：残余ネットワークの性質

残余ネットワークの閉路に注目



閉路の容量 α
＝閉路に含まれる枝の容量の最小値=1
閉路の費用 γ
＝閉路に含まれる枝の費用の和=-5

復習：残余ネットワークの性質

定理１：残余ネットワークに費用が負の閉路が存在
⇒　フローの費用を減少させることが可能
⇒　 現在のフローは費用最小でない

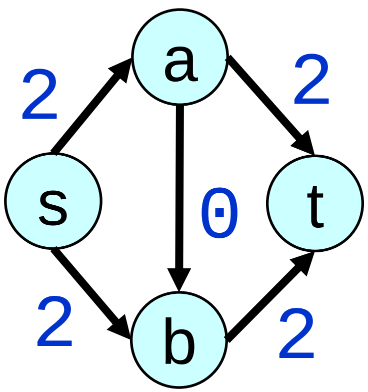
実は、逆も成り立つ（証明は後で）

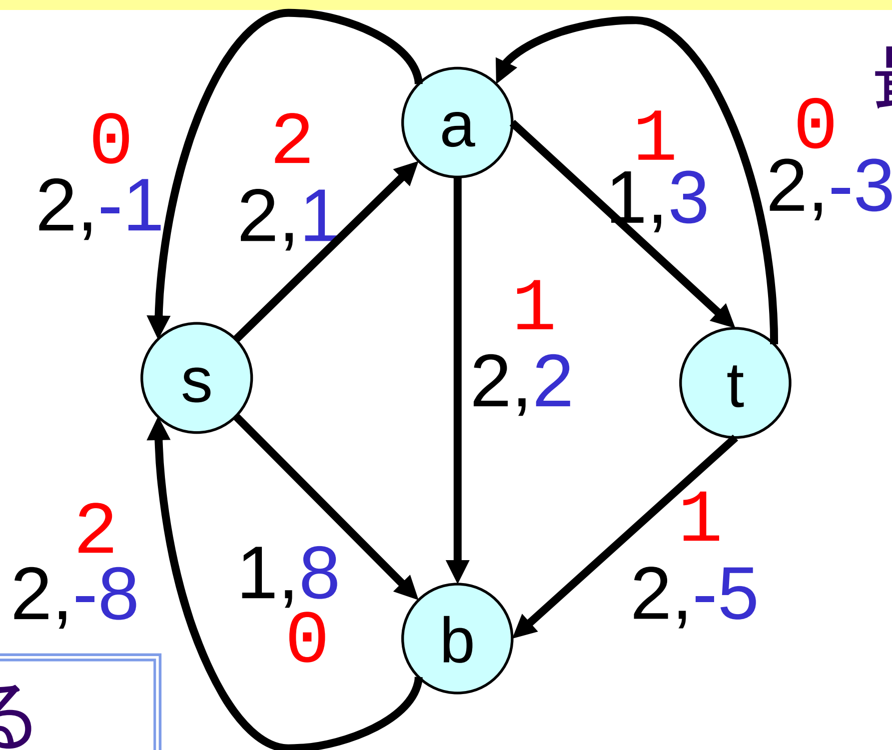
定理２：現在のフローは費用最小でない
⇒　残余ネットワークに費用が負の閉路が存在

定理2の証明(その1)

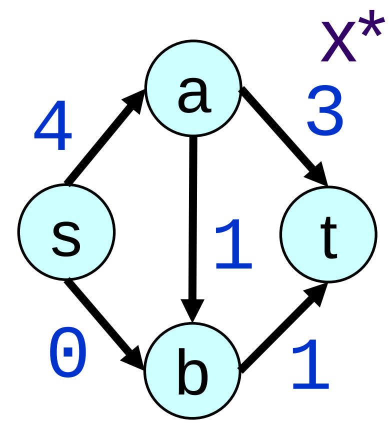
現在のフローは費用最小でない
⇒　残余ネットワークに費用が負の閉路が存在

現在のフロー x





最小費用フロー



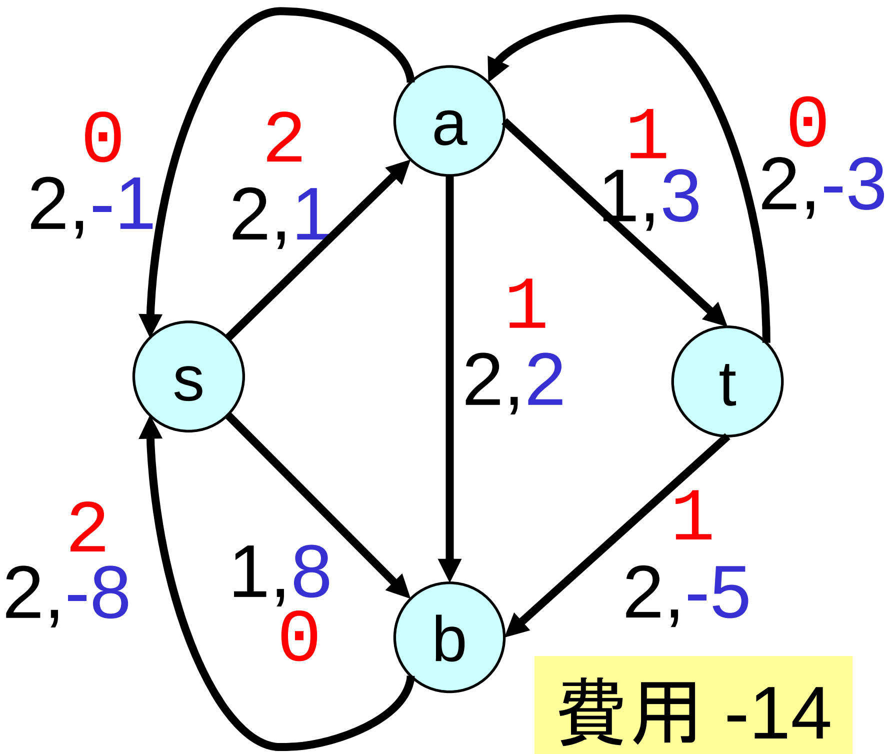
フロー x に関する
残余ネットワーク上で
x* と x の差を表現

$x_{ij}^* - x_{ij} \geqq 0 \Rightarrow$ 枝 (i, j) に $x_{ij}^*-x_{ij}$

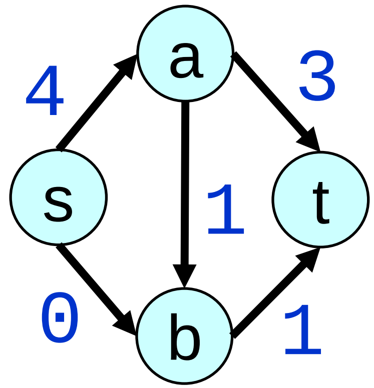
$x_{ij}^* - x_{ij} < 0 \Rightarrow$ 枝 (j, i) に $x_{ij} - x_{ij}^*$

その他の枝に0

定理2の証明(その2)

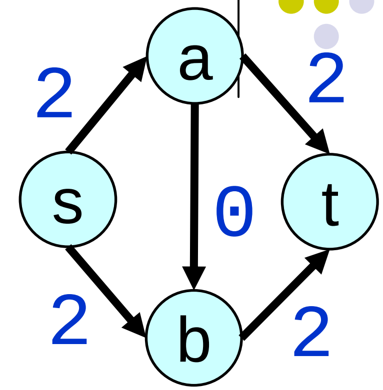


費用 -14



最小費用
フロー x*

費用 20



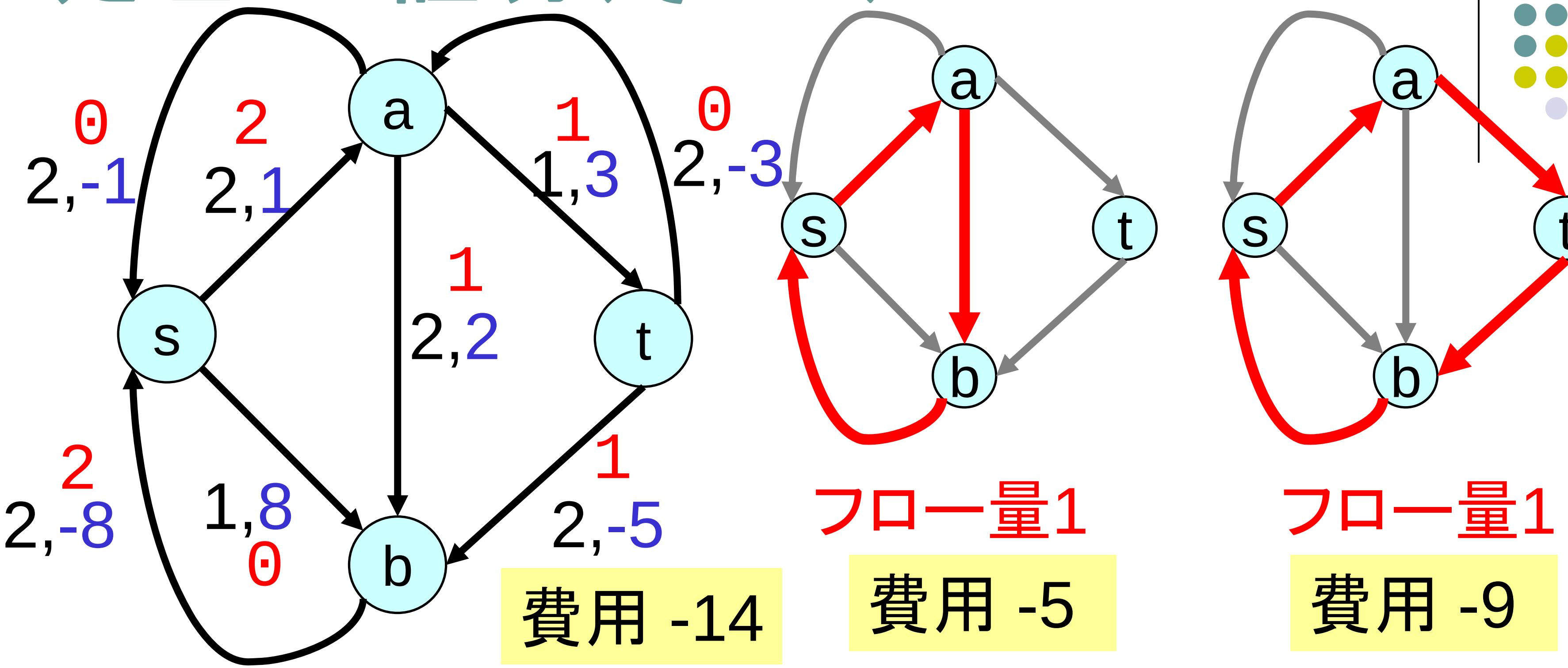
現在のフロー x

費用 34

x* と x の差：
x に関する残余ネットワーク上で
容量条件と流量保存則を満たす
⇒ 残余ネットワーク上の
「フロー」

差を表す「フロー」の費用
＝（x* の費用）
－（x の費用）
＜ 0

定理2の証明(その3)



x* と x の差を表す「フロー」:
x に関する残余ネットワーク上で
容量条件と流量保存則を満たす

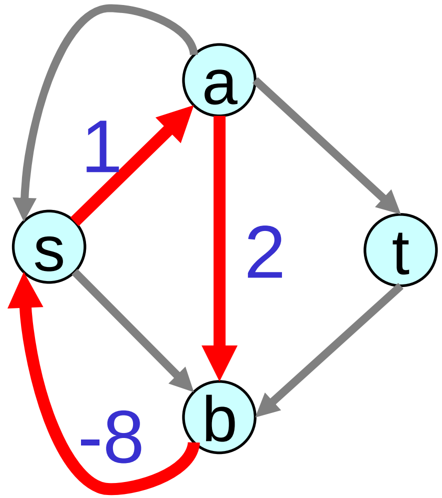
⇒ 閉路上を流れるフローに分解可能

差を表す「フロー」の費用
＝閉路上のフローの費用の合計

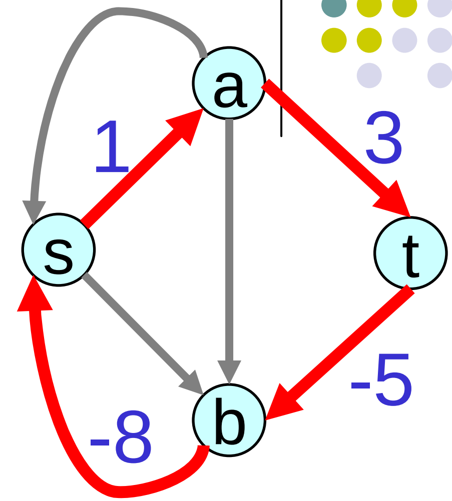
定理2の証明(その4)

閉路上のフローの
　費用の合計　費用 -14
＝差を表す「フロー」の費用
＜ 0

閉路上を流れるフロー
の費用
＝(閉路の費用)
　　　×(フロー量)



フロー量1
費用 -5



フロー量1
費用 -9

⇒ いずれかの閉路の費用は負

⇒ x の残余ネットワークに費用が負の閉路が存在
(証明終わり)

負閉路消去法の計算時間

※各枝の容量，費用は整数と仮定

U = 枝容量の最大値，

C = 枝費用の絶対値の最大値

m = 枝の数，n = 頂点の数

- 各反復においてフローの費用が1以上減る
- −mCU ≦ フローの費用 ≦ mCU

➔ 反復回数 ≦ 2mCU

- 各反復での計算時間

＝ 残余ネットワークの負閉路を求める時間

➔最短路問題のアルゴリズムを使うと O(m n) 時間

∴ 計算時間は $O(m^2 n C U)$

（入力サイズは m + n + log U + log C なので，指数時間）

負閉路消去法の改良

負閉路消去法の反復回数を少なくしたい
　➔ 各反復での負閉路の選び方を工夫する

（改良法１）費用減少量最大の負閉路を選ぶ
反復回数 O(m log (n U))
ただし，費用減少量最大の負閉路を求めるのはNP困難
➔ 費用減少量が最大に近い負閉路で代用可能

（改良法２）
“（閉路の費用）／（閉路の枝数）”が最小の負閉路を選ぶ
反復回数 $O(n\, m^2 \log n)$，一回の反復 O(n m)

※この他にも，負閉路消去法の計算時間を短縮するための様々なテクニックが存在する

最小費用フロー問題の応用例:研究室配属問題

- 各研究室に学生数人を割り当てる

学生A,B,C,Dの４人を研究室X,Yへ

- 各研究室に配属できる人数には上限がある

	X研究室	Y研究室
定員	3	3

- 学生の満足度の合計を最大にしたい

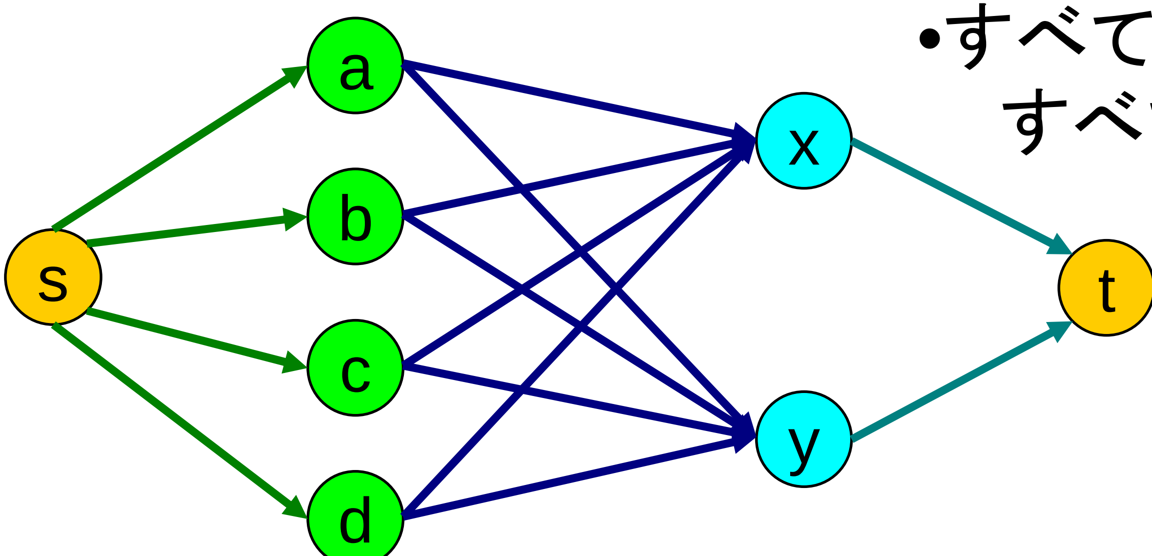
満足度	A	B	C	D
X	6	8	5	9
Y	9	1	5	3

応用例：研究室配属問題

最小費用フロー問題に変形

- 各学生に対応する頂点 a, b, c, d
- 各研究室に対応する頂点 x, y
- 供給点 s, 需要点 t

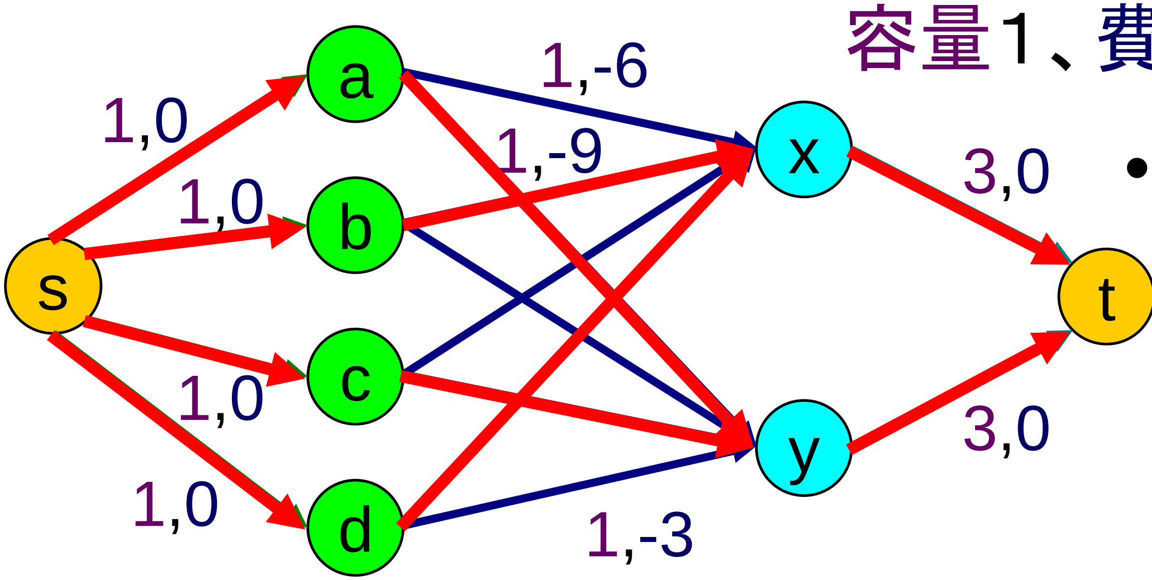
- 供給点から学生頂点への枝 (s, a), (s, b), …
- 研究室頂点から需要点への枝 (x, t), (y, t)



- すべての学生頂点から
すべての研究室頂点への枝
(a, x), (a, y), (b, x), …

応用例：研究室配属問題

•供給点から学生頂点への枝―容量1、費用0
•研究室頂点から需要点への枝
―容量＝研究室の定員、費用0
•学生頂点から研究室頂点への枝
容量1、費用＝（－1）×満足度



•需要（供給）量＝学生数

学生B,D→X研究室

学生A,C→Y研究室

この問題の（整数値）フロー⇔定員を満たす配属方法
フローの費用⇔（－1）×学生の満足度の合計
∴最小費用フロー問題に変形できた

非線形計画問題とは？

目的関数や制約式が**必ずしも線形でない**

数理計画問題

例：長方形の外周最小化問題

最小化　$2x + 2y$

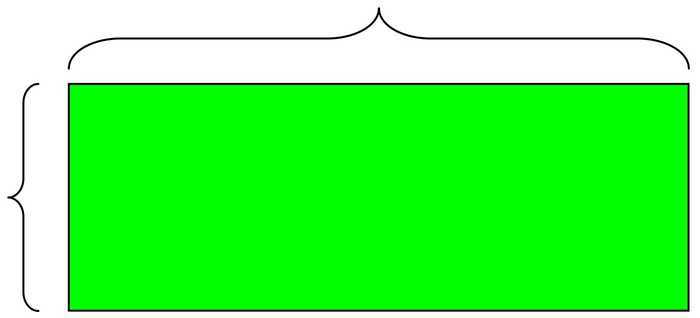
条件　$x\,y \geqq 1$

　　　$x, y \geqq 0$

非線形の不等式

y：横の長さ

x：縦の長さ

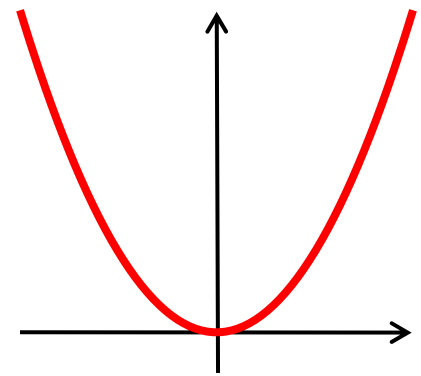


⇒ 非線形計画問題

注意:線形計画問題は非線形計画問題の特殊ケース

非線形関数の例（その1） [p.87]

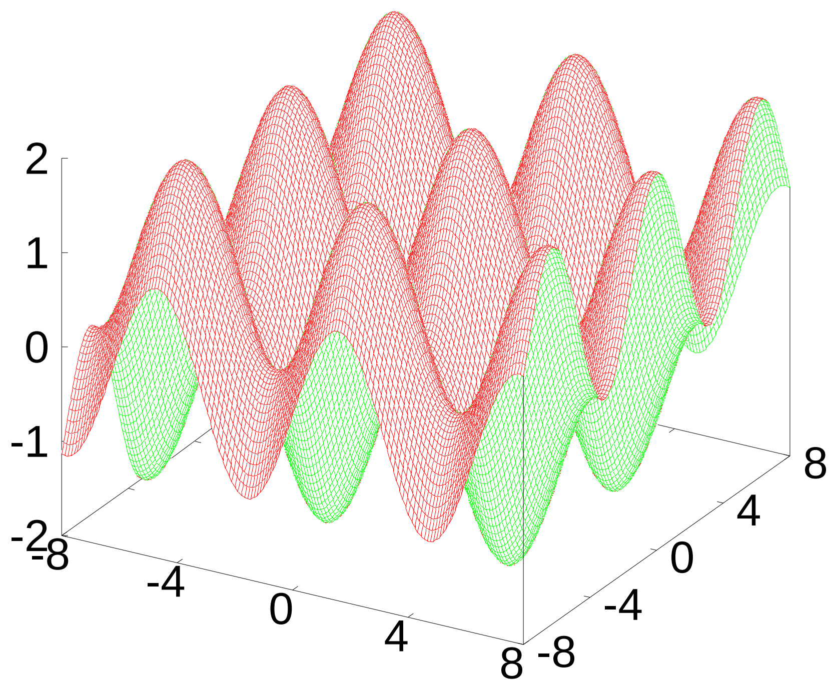
非線形関数　ー　線形でない関数



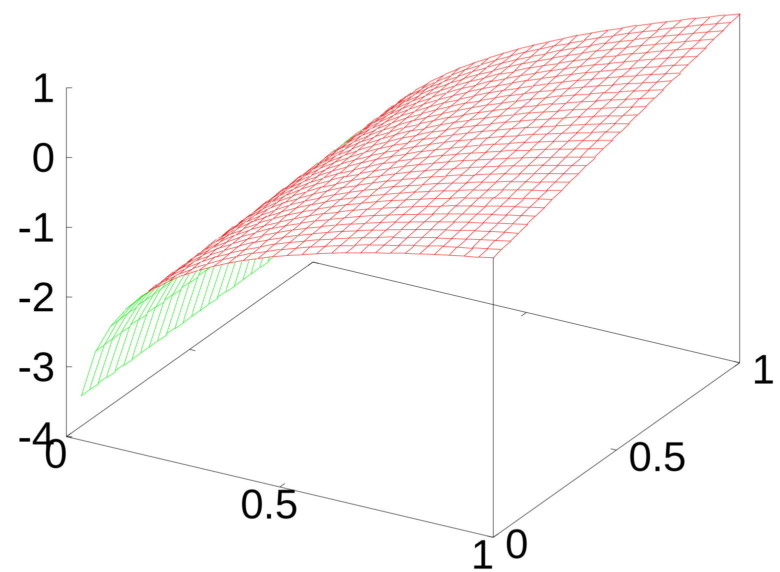
$$f_1(x) = x^2$$

微分可能な非線形関数の例

$$f_2(x_1, x_2) = \sin x_1 + \cos x_2$$



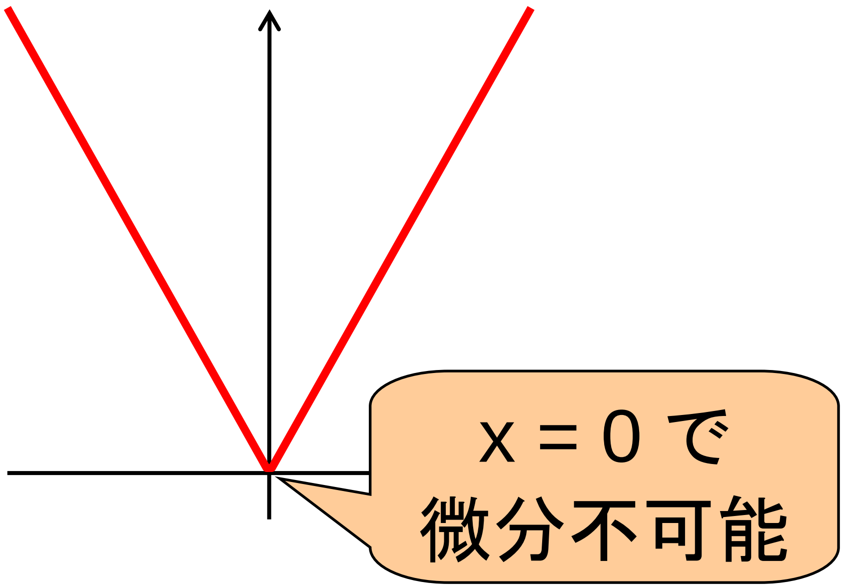
$$f_3(x_1, x_2) = x_1 x_2 + \log x_1$$



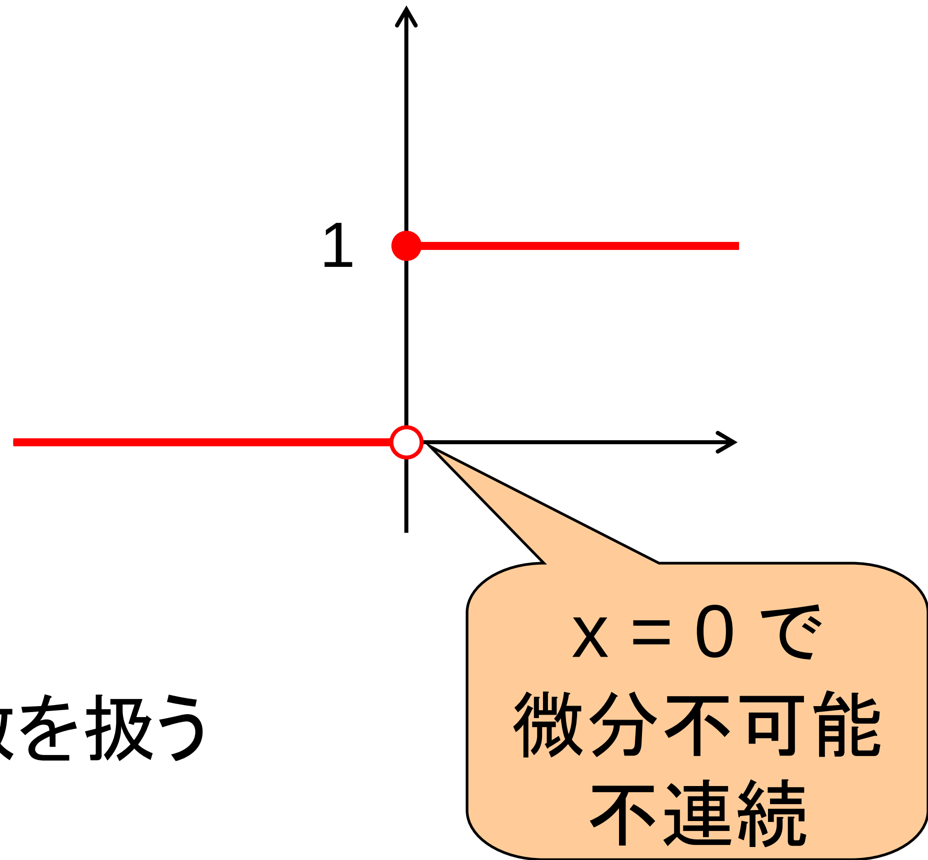
非線形関数の例(その2) [p.88]

微分不可能な非線形関数の例

$$f_4(x) = |x|$$



$$f_5(x) = \begin{cases} 1 & (x \geq 0\text{のとき}) \\ 0 & (x < 0\text{のとき}) \end{cases}$$



この授業:
主に何回でも微分可能な関数を扱う

非線形計画問題の分類 [p.6,97,131]

制約なし最適化問題

入力： 目的関数 $f(x)$
問題： 最小化 $f(x)$ 条件 なし

制約つき最適化問題

入力： 目的関数 $f(x)$, 制約を表す関数 $g_i(x)$ $(i = 1, 2, \ldots, m)$
問題： 最小化 $f(x)$ 条件 $g_i(x) \leqq 0$ $(i = 1, 2, \ldots, m)$

この講義では、制約なし問題を主に扱う

非線形計画問題の分類 [p.6,97,131]

制約つき問題と制約なし問題の関係

- 制約つき問題は制約なし問題に変形できる

最小化　f(x)　条件　$g_i(x) \leqq 0 \ (i = 1, 2, \ldots, m)$

⇒

最小化　f(x) + h(x)　条件　なし

$$h(x) = \begin{cases} 0 & (g_i(\boldsymbol{x}) \leq 0, i = 1,2,\ldots,m) \\ M & (\text{その他}) \end{cases}$$

Mは十分に大きい正数

（注意）この制約なし問題を直接解くことは実用上難しい

- 制約なし問題を繰り返し解くことにより、制約つき問題を解くことが出来る

→p.146 ペナルティ関数法、バリア関数法

勾配ベクトル [p.89]

関数 f の勾配ベクトル

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} \dfrac{\partial f}{\partial x_1} \\ \vdots \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{pmatrix}$$

一変数関数の場合は $\nabla f(x) = f'(x)$

例:

$$f_1(x) = x^2 \Rightarrow \nabla f_1(x) = 2x$$

$$f_2(x_1, x_2) = \sin x_1 + \cos x_2$$

$$\Rightarrow \nabla f_2(\boldsymbol{x}) = \begin{pmatrix} \cos x_1 \\ -\sin x_2 \end{pmatrix}$$

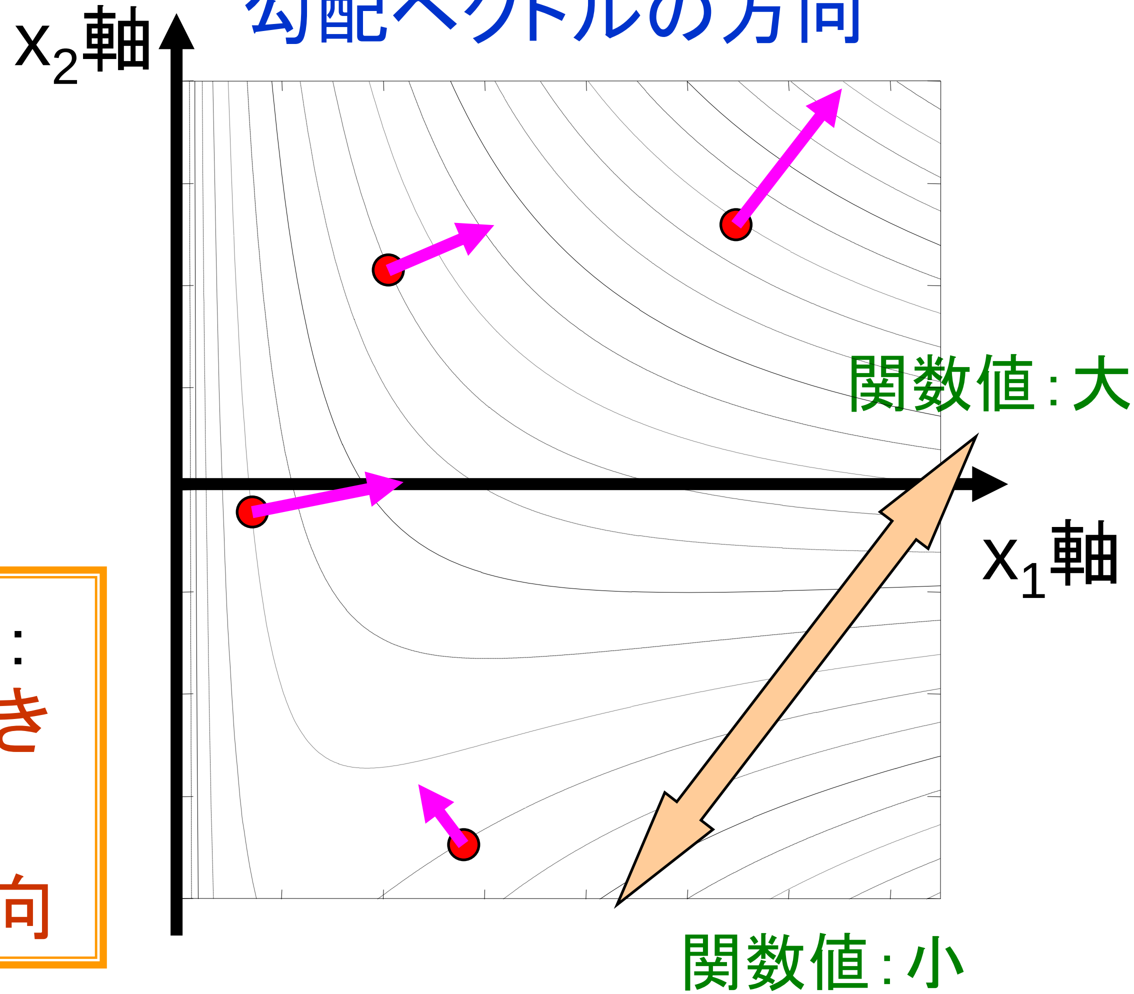
勾配ベクトル（続き） [p.89]

$$f_3(x_1, x_2) = x_1 x_2 + \log x_1$$

⇩

$$\nabla f_3(\boldsymbol{x}) = \begin{pmatrix} x_2 + 1/x_1 \\ x_1 \end{pmatrix}$$

関数f_3の等高線と
勾配ベクトルの方向



勾配ベクトルのイメージ：
- ■関数という山を登るときに最も急な方向
- ■関数値が増加する方向

一次のテイラー展開[p.89]

任意の関数 f は次の形に
表現できる（a は定数ベクトル）

関数 f(x) の x=a における
一次のテイラー展開

$$f(x) = \boxed{f(a) + \nabla f(a)^T (x-a)} + \varphi(x-a)$$

関数 $\varphi(x-a) = \varphi(x_1 - a_1, x_2 - a_2, \ldots, x_n - a_n)$ は
$x_1 - a_1, x_2 - a_2, \ldots, x_n - a_n$ に関する
２次以上の項からなる n 変数多項式関数
(定数項, 一次の項は全く含まれない)

$$f(a) + \nabla f(a)^T (x-a)$$

関数 f(x) の x=a における
一次のテイラー近似

- 線形関数
- x = a のとき値は f(a)
- 傾きは勾配ベクトル∇f(a)
- x=aの近くで関数 f を近似

一次のテイラー展開

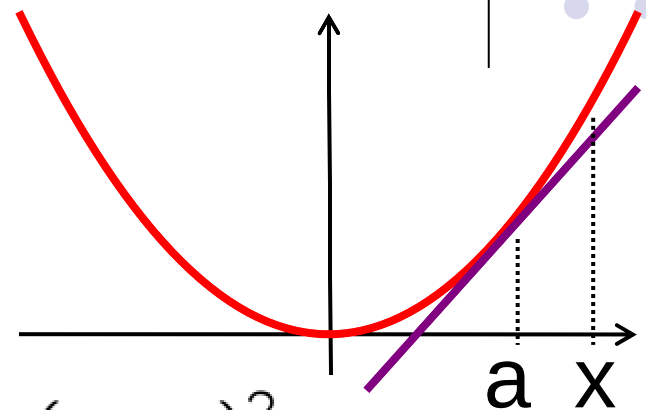
例1：$f_1(x) = x^2 \qquad f_1'(x) = 2x,$

f_1 の x=a における一次のテイラー展開

$$f(x) = \boxed{a^2 + 2a(x-a)} + \varphi(x-a)$$

ここで

$$\varphi(x-a) = f(x) - \{a^2 + 2a(x-a)\} = (x-a)^2$$



例2：$f_2(x) = \log x \qquad f_2'(x) = 1/x$

f_2 の x=1 における一次のテイラー展開

$$f_2(x) = \boxed{0 + \tfrac{1}{1}(x-1)} + \varphi(x-1)$$

ここで

$$\varphi(x-1) = -\frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \cdots$$