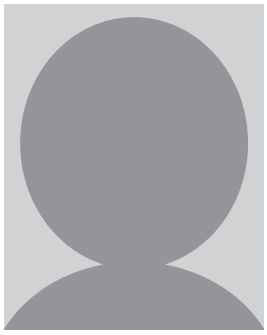
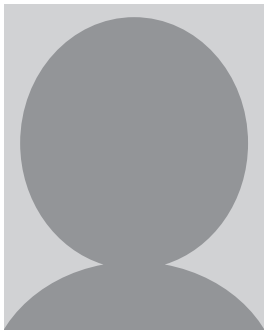
問題解決の数理（’１７）

- 収録本番とは多少異なっていることがあります.
- 内容の間違いのご指摘は歓迎します.
- 「完全に無保証」です.



- 今回の講義では，ネットワーク最適化法についてお話します．
- ネットワーク最適化問題は，ネットワーク構造を持つ対象・事象における，特定の目的に対する最適解を求める問題のことです．
- 現実世界には，交通網，通信網，人間関係，など，ネットワーク構造を持つシステムは多く存在し，目的地への最短経路を発見する等，ネットワーク上での最適化問題も数多く存在します．

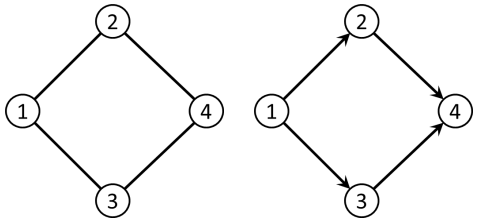


- 今回は，代表的なネットワーク最適化問題と，その定式化および解法についてお話します．
- まずは，ネットワークの数学的な表現法として，グラフについてお話します．

ネットワーク

グラフ

- 点と枝により構成
- 方向性の有無



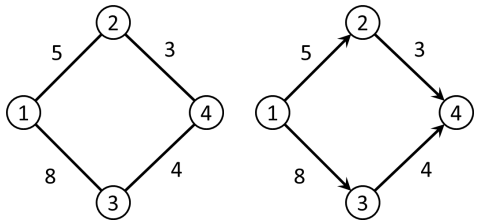
無向グラフ　　有向グラフ

- ここでいうグラフとは，グラフ理論の定義に基づくグラフです．
- グラフは点，それから点と点を結ぶ枝で構成されます．
- 点で事象や対象を表し，枝で点と点の間の関係を表すことにより，ネットワーク構造を持つシステムを表現することができます．
- グラフには方向性のない無向グラフと方向性のある有向グラフがあります．
- ここに無向グラフと有向グラフの例を示します．
- 有向グラフは枝を矢印にして方向性を表します．

ネットワーク

重み付きグラフ

- グラフには重みを与えることができる



無向グラフ　　有向グラフ

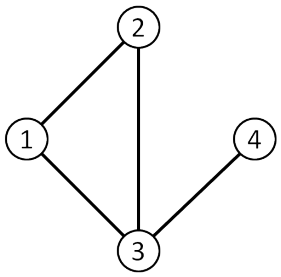
- グラフの枝には重みと呼ばれる数値を付与することができます.
- 重みは，点の間の距離など様々な事柄を表現できますが，重みの例は後ほど示します.

ネットワーク

グラフ

- 点 i と点 j を結ぶ枝 (i, j)
- 点の集合 V，枝の集合 E
- グラフ $G = (V, E)$

$$V = \{1, 2, 3, 4\},$$
$$E = \{(1, 2), (1, 3), (2, 3), (3, 4)\}$$



- ここまでグラフを図で表現しましたが，グラフを数学的に扱うためには次のようにします．
- 点 i と点 j を結ぶ枝を i, j と表します．
- 点の集合を V とします．この例では，V は，点 1, 2, 3, 4 からなります．
- 枝の集合を E とします．この例では，E は，(1, 2), (1, 3), (2, 3), (3, 4) からなります．
- グラフ G は，点の集合 V と枝の集合 E の組という形で表されます．
- この図の例のように，グラフは図の代わりに，V と E を用いて表わすことができます．
- 以降，いちいち断ることなく，G をグラフ，V を点の集合，E を枝の集合とします．
-

ネットワーク

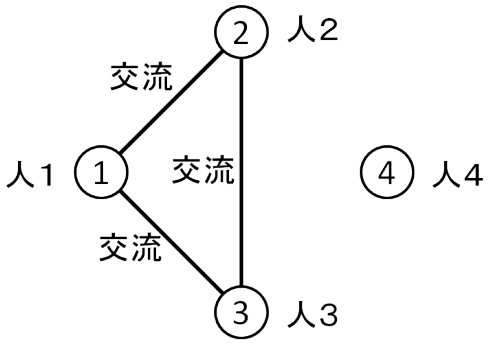
グラフネットワーク

- グラフによりネットワーク構造を持つシステムを表現
- 点は対象や事象
- 枝は点間の接続関係

- いま説明したグラフを用いて，ネットワーク構造を持つシステムを表現することができます.
- 一般的に，対象や事象を点で表し，点と点の関係を枝で表します.
- 簡単な例を示します.

ネットワーク

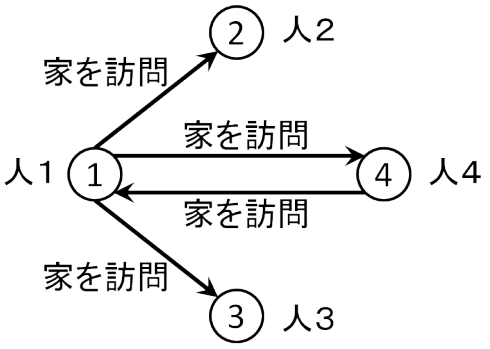
無向グラフによるネットワーク表現



- 無向グラフによる表現の例として，点で人，枝で交流の有無...，例えば，過去 1 年間に 10 度以上会ったことがあることを交流の定義とします．
- そうしますと，点と点が枝で結ばれていると，点に対応する「人と人」の間に交流があるということになります．
- 交流の定義から，交流に方向性はありませんので，無向グラフで表現するのが適切でしょう．

ネットワーク

有向グラフによるネットワーク表現



- 有向グラフによる表現の例として，例えば点で人，枝で家の訪問を表現するとします．
- 例えば，1 番の人は 2 番の人の家を訪問していますが，2 番の人は 1 番の人の家を訪問していません．
- 1 番の人と 4 番の人は互いに家を訪問し合っています．
- 家の訪問には，明らかに方向性がありますので，有向グラフで表現するのが適切でしょう．

ネットワーク

交通網

- 点で地点（交差点，駅）
- 枝で道路，運行
- 枝の重みで
  - 地点間の距離
  - 地点間の所要時間
  - 地点間の料金

- 次に，重み付きのネットワークの例を紹介します．
- 例えば，道路網や鉄道網といった交通網は，点で交差点や駅などの地点を表し，枝で地点と地点の接続，例えば道路や列車の運行を表します．
- そして，重みで地点間の距離，所要時間，所要料金などを表します．

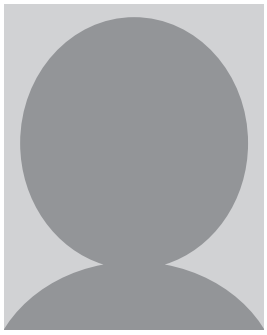
ネットワーク

情報通信

- 点で端末
- 枝で接続
- 重みで
  - 帯域幅
  - コスト
  - 遅延

- 通信ネットワークでは，点でPCやスマートフォンなどの端末やルータなどの中継器を表し，枝でケーブルや無線での物理的接続や論理的接続を表します．
- そして，重みで端末間で使用できる帯域幅，使用コスト，通信遅延などを表します．
- このようなネットワーク構造をもつシステムに関する最適化問題がネットワーク最適化問題です．

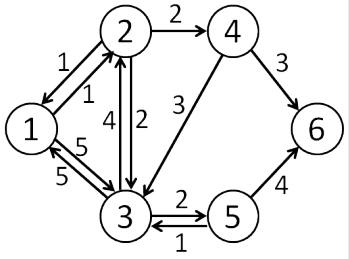
最短路問題



- ここでは，代表的なネットワーク最適化問題である最短路問題とその定式化を示します.

最短路問題

- 点は地点，枝は道路，重みは枝で結ばれた地点間の距離
- 地点 1 から地点 5 に最短で移動する経路を求める



- 最短路問題とは次のような問題です．
- 図において，点は地点，枝は道路，枝にふられた数字が重みですが，重みは地点間の距離を表しています．
- 点 1 から点 5 への最短経路を求めよ…という問題です．

最短路問題

最短路問題の応用

- カーナビゲーション・システム
  - 最短経路の発見
- 鉄道路線の乗り換え案内
  - 所要時間最短
  - 運賃最安
  - 乗換回数最少
  - 通信ネットワークの経路制御

- 最短路問題は，カーナビゲーション・システムにおける最短路の発見，鉄道路線の乗り換え案内における，所要時間が最短の経路，運賃が最も安い経路，乗換回数が最少の経路の発見，通信ネットワークにおける経路制御など多くの問題に当てはまります．
- それでは，最短路問題を定式化していきます．
- まずは，一般的な形で定式化していきます．

最短路問題

決定変数

- 枝 (i, j): 地点 i と地点 j を結ぶ道路
- 決定変数 x_{ij}: 枝 $(i, j) \in E$ を最短路に含めるか否か
- (i, j) を最短路に

$$\begin{array}{rl} \text{含める} & x_{ij} = 1 \\ \text{含めない} & x_{ij} = 0 \end{array}$$

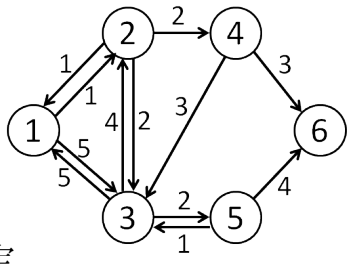
- まず，決定変数を定義します．
- 地点 i と地点 j を結ぶ道路 (i, j)，すなわち，点 i と点 j を結ぶ枝 (i, j) を x_{ij} であらわします．
- 枝 (i, j) が最短路に含まれる時 $x_{ij} = 1$ とし，枝 (i, j) が最短路に含まれない時 $x_{ij} = 0$ とします．

最短路問題

目的関数の定式化

- w_{ij}: (i,j) 間の距離
- z: 移動距離

$$z = \sum_{(i,j)\in E} w_{ij}x_{ij}$$



- z を最小化する x_{ij} の値を決定

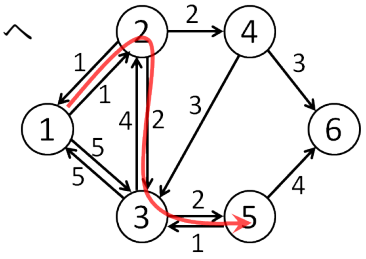
- 枝 (i,j) の重みを w_{ij} として，w_{ij} は地点 i と地点 j の距離とします．
- 最短路を求める問題ですから，目的関数 z は，すべての枝 (i,j) に対する，w_{ij} かける x_{ij} の総和とします．
- x_{ij} は最短路に含まれる枝のみが 1 で，残りは 0 になりますので，正しく最短路が求められた時，z の値は最短路の距離になります．
- つぎに，制約条件を定式化していきます．

最短路問題

始点（出発地点）の制約の定式化

- 出発地点 s では，他の 1 地点へ出るだけで，
  他の地点からは入らない

$$\sum_{(s,j)\in E} x_{sj} - \sum_{(j,s)\in E} x_{js} = 1$$



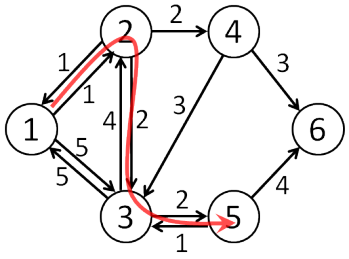
- 先に最短路を示しておきます．
- 出発地点が点 1，到着地点が点 5 で，最短路は赤い矢印で示された経路，すなわち $1 \to 2 \to 3 \to 5$ が最短路です．
- まず，出発地点 s に注目します．
- 一般に出発地点 s は複数の地点と繋がっているので，枝は複数あります．
- この例では，出発地点は点 1 ですが，点 2 と点 3 に繋がる枝があります．
- 最短路は 1 本だけですから，出発地点から出る枝のうち，最短路に含まれるのは 1 本だけです．すなわち，各点を j とすると，x_{sj} のうち，値が 1 となるのは 1 個だけで，残りは 0 となります．したがって，x_{sj} の総和は 1 となります．
- 一方，最短路は出発地点に入ることはありませんので，x_{js} はすべて 0 となります．したがって，x_{js} の総和は 0 となります．
- これらをまとめて，x_{sj} の総和から x_{js} の総和を引くと 1 となります．

最短路問題

終点（到着地点）の制約の定式化

- 到着地点 t では，
  他の 1 地点から入るだけで，
  他の地点に出ない

$$\sum_{(t,j)\in E} x_{tj} - \sum_{(j,t)\in E} x_{jt} = -1$$



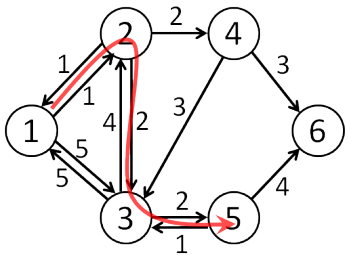
- 次に，到着地点 t に注目します．
- 到着地点 t も一般には複数の地点と繋がっているので，枝は複数あります．
- この例では，到着地点は点 5 ですが，点 3 と点 6 に繋がる枝があります．
- 最短路は 1 本だけですから，到着地点に入る枝のうち，最短路に含まれるのは 1 本だけです．すなわち，各点を j とすると，x_{jt} のうち，値が 1 となるのは 1 個だけで，残りは 0 となります．したがって，x_{jt} の総和は 1 となります．
- 一方，最短路は到着地点から出ることはありませんので，x_{tj} はすべて 0 となります．したがって，x_{tj} の総和は 0 となります．
- これらをまとめて，x_{tj} の総和から x_{jt} の総和を引くと -1 となります．

最短路問題

出発地点と到着地点以外の地点の制約の定式化

- 各地点 i (s, t 以外)
- 他の 1 地点から入って
  別の 1 地点に出る
- その地点を通らない

$$\sum_{(i,j)\in E} x_{ij} - \sum_{(j,i)\in E} x_{ji} = 0$$



- 最後に，出発地点 s，到着地点 t 以外の地点に注目します．
- まず，その地点が最短路に含まれる場合を考えます．
- 例えば地点 2 は最短路に含まれていますが，最短路においては，地点 2 には地点 1 から入り，地点 2 からは地点 3 へ出ています．
- すなわち，地点 2 の出入りは共に 1 つです．
- 今度は，その地点が最短路に含まれない場合を考えます．
- 例えば地点 4 は最短路に含まれていません．
- したがって，地点 4 の出入りはありません．
- 注目する地点を i としますと，地点 i が最短路に含まれる時，x_{ij} の総和および x_{ji} の総和は 1 です．
- 地点 i が最短路に含まれない時，x_{ij} の総和および x_{ji} の総和は 0 です．
- これらをまとめて表現すると，x_{ij} の総和から x_{ji} の総和を引くと 0，となります．

最短路問題

最小化	$z = \sum_{(i,j)\in E} w_{ij}x_{ij}$	移動距離
制約条件	$\sum_{(s,j)\in E} x_{sj} - \sum_{(j,s)\in E} x_{js} = 1$	始点 s の出入
	$\sum_{(t,j)\in E} x_{tj} - \sum_{(j,t)\in E} x_{jt} = -1$	終点 t の出入
	$\sum_{(i,j)\in E} x_{ij} - \sum_{(j,i)\in E} x_{ji} = 0$	その他地点の出入
	$x_{ij} \in \{0,1\}$ for $(i,j) \in E$	x_{ij}は 0 か 1

- 以上をまとめると，最短路問題は次のように定式化されます.
- ここで注目していただきたいのは，決定変数 x_{ij} が 0 と 1 の 2 値のみをとることです.
- この問題は線形最適化問題ではなく，整数最適化問題，より正確には「0-1 整数最適化問題」と呼ばれる問題になります.
- 0-1 整数最適化問題は組み合わせ最適化問題の一種で，線形最適化問題と比べ計算量ははるかに大きくなります.
- しかし幸いなことに，重み w_{ij} がすべて非負整数であれば，この構造の問題は決定変数が 0 か 1 の制約を外して、線形最適化問題として定式化しても同じ最適解が得られることが知られています.

最短路問題

最小化	$z = \sum_{(i,j)\in E} w_{ij}x_{ij}$	移動距離
制約条件	$\sum_{(s,j)\in E} x_{sj} - \sum_{(j,s)\in E} x_{js} = 1$	始点 s の出入
	$\sum_{(t,j)\in E} x_{tj} - \sum_{(j,t)\in E} x_{jt} = -1$	終点 t の出入
	$\sum_{(i,j)\in E} x_{ij} - \sum_{(j,i)\in E} x_{ji} = 0$	その他地点の出入
	$x_{ij} \geq 0$ for $(i,j) \in E$	x_{ij}の非負条件

- この性質を利用すると，最短路問題はこのような線形最適化問題として定式化されます.
- 整数最適化問題からの変更は，x_{ij} が 0 か 1 の 2 値をとるという制約が，非負の数と緩められていることです.
- ここで，$x_{ij} \leq 1$ という条件がない... ということが気になるかと思いますが，この条件はなくても同じ最適解が得られます.

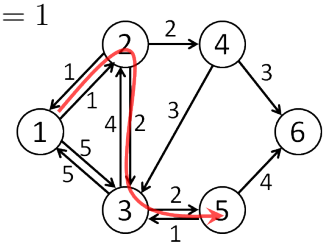
最短路問題

最小化 $z = x_{12} + 5x_{13} + x_{21} + 2x_{23} + 2x_{24}$
$+ 5x_{31} + 4x_{32} + 2x_{35}$
$+ 3x_{43} + 3x_{46} + x_{53} + 4x_{56}$

制約条件

始点 (点 1) $(x_{12} + x_{13}) - (x_{21} + x_{31}) = 1$

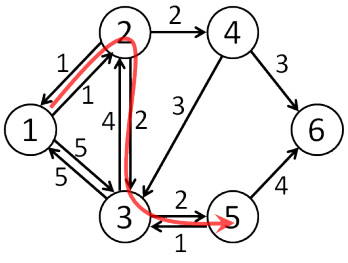
終点 (点 5) $(x_{53} + x_{56}) - x_{35} = -1$



- 一般的な定式化ができましたので，今度はより具体的に定式化します．
- この問題において，点は 1 から 6，枝はグラフに示す通りです．$w_{ij} \times x_{ij}$ の総和が目的関数ですので，目的関数 z はこの式のようになります．
- 次に各点における制約です．
- 始点は点 1 で，点 1 は点 2 と点 3 につながっています．
- 最短路は 1 本だけですから，点 1 から出る枝のうち，最短路に含まれるのは 1 本だけです．すなわち，
  $x_{12} + x_{13} = 1$ となります．
- 一方，最短路は点 1 に入ることはありませんので，
  $x_{21} + x_{31} = 0$ となります．
- これらを合わせると，始点，点 1 における制約はこの式になります．
- 終点は点 5 で，点 5 には点 3 から入る枝があります．
- 最短路は 1 本ですから，$x_{35} = 1$ となります．
- 一方，点 5 から点 3 と点 6 に出る枝がありますが，最短路は点 5 から出ることはありませんので，$x_{53} + x_{56} = 0$ となります．
- これらを合わせて，終点，点 5 における制約はこの式になります．

最短路問題

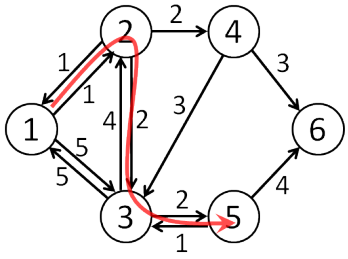
点 2　$(x_{21} + x_{23} + x_{24}) - (x_{12} + x_{32}) = 0$
点 3　$(x_{31} + x_{32} + x_{35}) - (x_{13} + x_{23} + x_{43} + x_{53}) = 0$
点 4　$(x_{43} + x_{46}) - x_{24} = 0$
点 6　$-(x_{46} + x_{56}) = 0$
$x_{ij} \geq 0$ for $(i, j) \in E$



- 次に，始点と終点以外の点における制約です．
- 点 2 は，点 1，点 3，点 4 に出ていく枝があり，点 1，点 3 から入ってくる枝がありますので，点 2 における制約は，この式のようになります．
- 点 3，点 4，点 6 における制約も同様ですので，確認しておいて下さい．

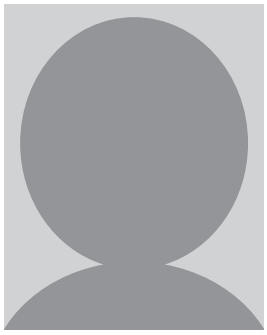
最短路問題

- この問題の最適解
  $x_{12}, x_{23}, x_{35} = 1$，他の $x_{ij} = 0$
- 経路 $1 \to 2 \to 3 \to 5$ が最短路

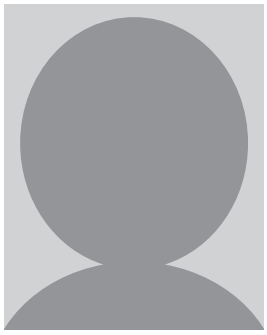


- この問題の最適は既に示してありますが，x_{12}, x_{23}, x_{35} が 1 で，その他の x_{ij} は 0 です．
- すなわち，最短路は点 1 から点 2，点 3，点 5 と移動する経路で，距離は $1 + 2 + 2 = 5$ となります．

ダイクストラ法



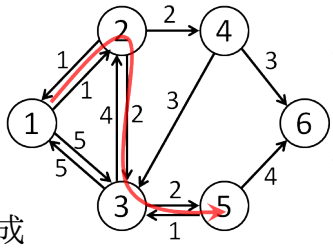
- 最短路問題を，整数最適化問題および線形最適化問題として定式化しました．
- こうして定式化できましたので，整数最適化法や線形最適化法の汎用的解法，例えば整数最適化法なら分枝限定法など，線形最適化法ならシンプレックス法などを用いれば，最短路問題を解くことができます．
- しかし，最短路問題は問題の構造を利用して，さらに効率的に解くことができます．



- 今回紹介するダイクストラ法は，最短路問題を効率的に解く解法の一つで，カーナビゲーション・システムにおける経路探索などに広く用いられています．
- また，アルゴリズムも単純ですので，ここで紹介します．
-

ダイクストラ法

- 最短路問題を効率良く解くアルゴリズム
- 最短路
  $1 \to 2 \to 3 \to 5$
  の部分もその
  区間の最短路
- 始点から始点に
  近い地点までの
  最短路を順次構成



$\underbrace{1 \to 2 \to 3 \to 5}$
1 から 5 までの最短路

$\underbrace{1 \to 2 \to 3}$
1 から 3 までの最短路

$\underbrace{1 \to 2}$
1 から 2 までの最短路

- これは，先ほどの最短路問題の例ですが，点 1 から点 5 への最短路は，１，２，３，５という経路，です．
- ここで，注目してほしいのは，この最短路の部分経路，例えば 1，２，３は，点 1 から点 3 への最短路になっていることです．
- つまり，最短路問題においては，最短路のすべての部分経路は最短路になっています．
- 一般に，問題を部分問題に分解した時，元の問題の最適解が部分問題の最適解を含んでいるとき,「最適性の原理」が成り立っているといいます．
- 最適性の原理を利用して効率的に問題を解く方法を「動的計画法」といいます．
- ダイクストラ法は最短路問題における動的計画法で，始点に近い点への最短路から順次構成していく方法です．
- それでは，次にダイクストラ法のアルゴリズムを示します．
- ここでは，始点を 1 として，始点 1 から他のすべての点への最短路を求めることにします．
- また，枝の重み…距離のことですが，重みの値は非負であるとします．
-

ダイクストラ法

0. 初期化
   $S \leftarrow \{\}$, $N \leftarrow V$, $d(s) \leftarrow 0$,
   $d(i) \leftarrow \infty$ $(i \in V \backslash \{s\})$

S: 最短路が確定した点の集合
N: 最短路が確定しない点の集合
$(N = V \backslash S)$
$d(i)$: 始点 s から点 i までの暫定最短距離

- まず，用いる記号を説明します．
- S は最短路が確定した点の集合を表します．
- 一方，N は最短路が確定していない点の集合を表します．
- S と N は補集合の関係にあります．
- ここで，V バックスラッシュS は，V から S を除いた集合を表します．
- なお，印刷教材では N ではなく，S バーを用いていますが，画面上では見にくいので，代わりに N を使います．
- それから，$d(i)$ は始点 s から点 i までの最短距離の暫定値，上限値を表します．
- ダイクストラ法の最初のステップは初期化を行います．
- すべての点の最短路は確定していないので，S は空（くう），空っぽとし，N にすべての点を入れます．
- また，始点 s への最短距離の暫定値 $d(s)$ は 0 とし，始点 s 以外の点 i への最短距離の暫定値 $d(i)$ は分からないので，∞ としておきます．
-

ダイクストラ法

1. **if** $S = V$ **then** 終了
   else $d(v) \leftarrow \min\{d(i) | i \in N\}$ である点 v を選ぶ

- すべての点への最短路が確定したら $(S = V)$ 終了
- 未確定の点があれば，未確定の点の集合 N から，$d(i)$ が最小になる点を選ぶ (点 v とする)

- 次のステップでは，まず終了条件をチェックします.
- すべての点への最短路が確定したら，すなわち $S = V$ となれば，終了します.
- すべての点への最短路が確定していない場合，最短路の暫定値 $d(i)$ の最も小さい点 i を選び，これを v とします.
- これで，点 v までの最短路が確定します.
- なぜなら，距離は非負ですから，他の未確定の点を経由しても，点 v までの距離が $d(v)$ より短くなることがないからです.

ダイクストラ法

2. $S \leftarrow S \cup \{v\}$, $N \leftarrow N \backslash \{v\}$
   $(v, j) \in E$ かつ $j \in N$ である枝 (v, j) に対して，
   $d(j) > d(v) + w_{vj}$ ならば $d(j) \leftarrow d(v) + w_{vj}$,
   $p(j) \leftarrow v$ として (1) に戻る

- v への最短路は確定→ v を N から S へ移す
- v から直接つながっている N 内の点に対して，
  v 経由の方が距離が短ければ，経路を v 経由に変更
  ※ $p(j)$: s から点 j までの最短路における j の直前の点

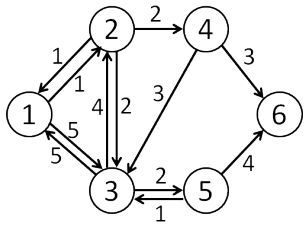
- そこで，次のステップでは，まず，最短路が確定した
  点 v を N から S へ移します．
- 次に，点 v から枝でつながっている N の点に対して，点 v を経由する方が，始点からの距離が短くなるならば，経路を「v 経由」に変更します．
- 最短距離の暫定値を更新するとともに，始点 s から点 j までの最短路における j の直前の点，$p(j)$ を v とします．
- $p(j)$ は，点 j から始点 s へ，最短路を逆向きに辿る手がかりになります．
- このステップが完了すると，ひとつ前のステップに戻り，すべての点までの最短路が確定するまで，これらのステップを繰り返します．
-
- 以上がダイクストラ法のアルゴリズムです．
- アルゴリズム自体は単純で，計算機プログラムを作るのも易しいのですが，最短路を構成する過程とイメージが結びつかないので，分かった気がしない，ということを複数の人から聞きました．
- 納得の手段として，一般論ですが，一つはアルゴリズムの正しさを数学的に証明してみる，あるいは証明を読んでみるということがあります．
- 証明を読んだけど分からない，証明はできたが分かった気がしないという場合には，アルゴリズムを具体的な問題に適用して，そのステップを確認するということも有効かもしれません．
- ここでは，ダイクストラ法の動作をより具体的に示すために，ダイクストラ法の適用例を示し，そのステップを追っていくことにします．

ダイクストラ法

（0）

$S \leftarrow \{\}$, $N \leftarrow \{1, 2, 3, 4, 5, 6\}$

$d(1) \leftarrow 0$,
$d(2), d(3), d(4), d(5), d(6) \leftarrow \infty$



- 問題はこれまでと同じものにします.
- 最初はどの点も最短路が確定していないので，S は空（くう）とし，N には点 1 ～ 6 のすべてを入れます.
- また，始点である点 1 に関しては，$d(1)$ は 0，始点以外の点に関しては d は ∞ とします.

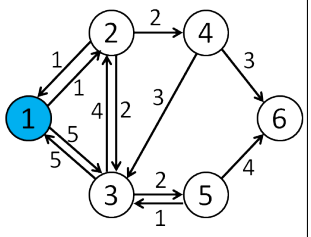
ダイクストラ法

1 回目（1）

$S = \{\}$, $N = \{1, 2, 3, 4, 5, 6\}$

$d(1) = 0$,
$d(2) = d(3) = d(4) = d(5) = d(6) = \infty$

$\min\{d(1), d(2), d(3), d(4), d(5), d(6)\}$
$= \min\{0, \infty, \infty, \infty, \infty, \infty\} = 0$
より $v = 1$

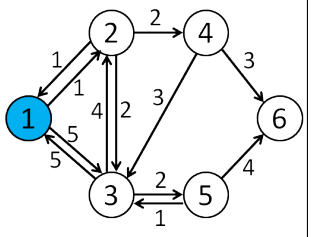


- 次のステップです.
- まず，$S = V$ ではないので，まだ終了しません.
- 始点からの距離 d が最も短いのは，点 1 ですので，点 1 までの最短路... といっても始点自身ですが，点 1 までの最短路は確定です.

ダイクストラ法

1 回目（2）

$S \leftarrow \{1\}$, $N \leftarrow \{2,3,4,5,6\}$



- 次のステップでは，まず，確定した点 1 を N から S に移します．

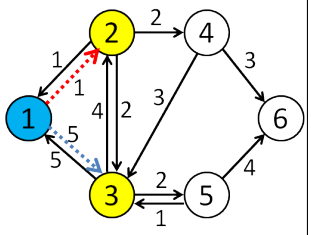
ダイクストラ法

1 回目（2）

$S = \{1\}$, $N = \{2, 3, 4, 5, 6\}$

$d(2) = \infty > d(1) + w_{12} = 0 + 1 = 1$
$\rightarrow d(2) \leftarrow 1$, $p(2) \leftarrow 1$

$d(3) = \infty > d(1) + w_{13} = 0 + 5 = 5$
$\rightarrow d(3) \leftarrow 5$, $p(3) \leftarrow 1$



- 次に，点 1 から枝で直接つながっている N の点，点 1 からは点 2 と点 3 が直接つながっています．
- まず，点 2 に関して，現在 $d(2) = \infty$ です．
- 点 1 から点 2 への距離 w_{12} は 1 です．
- $d(1) + w_{12} = 0 + 1 = 1$ は，$d(2)$ より短いので，$d(2)$ を 1 と更新して，点 2 までの暫定最短路を経路 1 ， 2 とします．
- 暫定最短路における点 2 の直前の点は 1 ですから，$p(2)$ を 1 とします．
- 次に，点 3 に関しては，現在 $d(3) = \infty$ で，点 1 から点 3 への距離 w_{13} は 5 です．
- $d(1) + w_{13} = 0 + 5 = 5$ は，$d(3)$ より短いので，$d(3)$ を 5 と更新して，点 3 までの暫定最短路を経路 1 ， 3 とします．
- 点 3 までの暫定最短路における，点 3 の直前の点は 1 ですから．$p(3)$ を 1 とします．
-

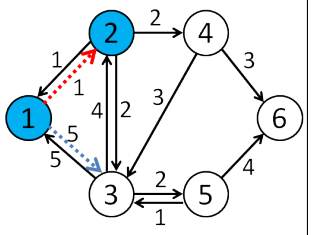
ダイクストラ法

2 回目（1）

$S = \{1\},\ N = \{2, 3, 4, 5, 6\}$

$d(2) = 1,\ d(3) = 5,$
$d(4) = d(5) = d(6) = \infty$

$\min\{d(2), d(3), d(4), d(5), d(6)\}$
$= \min\{1, 5, \infty, \infty, \infty\} = 1$
より $v = 2$

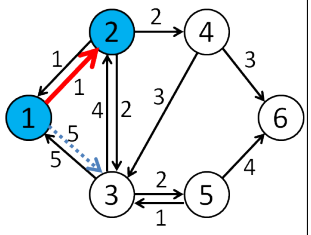


- ここから 2 回目の繰り返しです.
- まず，終了条件は満たしていないので，N の中から d が最短になる点 v を選びます.
- ここでは，$d(2)$ が 1 で最小なので，v は 2 となります.

ダイクストラ法

2回目（2）

$S \leftarrow \{1, 2\}$, $N \leftarrow \{3, 4, 5, 6\}$



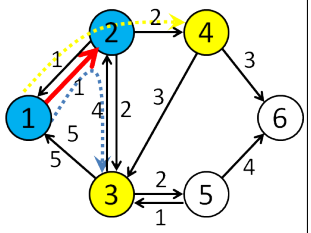
- ステップ (2) では，まず，確定した点 2 を N から S に移します．

ダイクストラ法

2 回目（2）

$S = \{1, 2\}$, $N = \{3, 4, 5, 6\}$

$d(3) = 5 > d(2) + w_{23} = 1 + 2 = 3$
$\rightarrow d(3) \leftarrow 3$, $p(3) \leftarrow 2$
$d(4) = \infty > d(2) + w_{24} = 1 + 2 = 3$
$\rightarrow d(4) \leftarrow 3$, $p(4) \leftarrow 2$



- 次に，点 2 から枝で直接つながっている N の点，点 2 からは点 3 と点 4 が直接つながっています．
- まず，点 3 に関して，暫定最短路は経路 1 ， 3 で，$d(3) = 5$ です．
- $d(2) = 1$ で，点 2 から点 3 への距離 w_{23} は 2 です．
- $d(2) + w_{23} = 1 + 2 = 3$ で $d(3)$ より短いので，$d(3)$ を 3 と更新して，点 3 までの暫定最短路を経路 1 ， 2 ， 3 に更新します．
- 暫定最短路における点 3 の直前の点は 2 ですから，$p(3)$ を 2 と更新します．
- 次に，点 4 に関しては，現在 $d(4) = \infty$ で，点 2 から点 4 への距離 w_{24} は 2 です．
- $d(2) + w_{24} = 1 + 2 = 3$ は $d(4)$ より短いので，$d(4)$ を 3 と更新して，点 4 までの暫定最短路を経路 1 ， 2 ， 4 とします．
- 点 4 までの暫定最短路における点 4 の直前の点は 2 ですから，$p(4)$ を 2 とします．

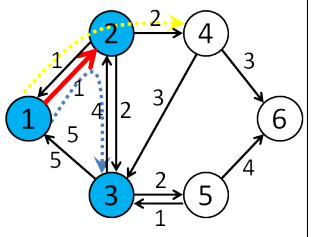
ダイクストラ法

3 回目（1）

$S = \{1, 2\}$, $N = \{3, 4, 5, 6\}$

$d(3) = 3, d(4) = 3,$
$d(5) = d(6) = \infty$

$\min\{d(3), d(4), d(5), d(6)\}$
$= \min\{3, 3, \infty, \infty\} = 3$
より $v = 3$ ($v = 4$ でもよい)

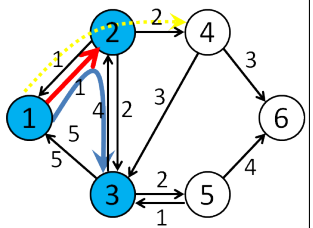


- ここから 3 回目の繰り返しです.
- まず，終了条件は満たしていないので，N の中から d が最短になる点 v を選びます.
- ここでは，$d(3)$ と $d(4)$ が 3 で最短です.
- 点 3 と点 4 のどちらを選んでもよいのですが，ここでは点 3 を選ぶことにします.
- なお，点 4 を選んだ場合でも最終的には同じ結果になります.
- ここで，点 4 を選んだ場合の探索のプロセスを追っている時間はありませんが，是非自分で探索のプロセスを追ってみて下さい.
- 話を元に戻しましょう.
- N の中から d が最短になる点 v として，点 3 を選びます.

ダイクストラ法

3回目（2）

$S \leftarrow \{1, 2, 3\}$, $N \leftarrow \{4, 5, 6\}$



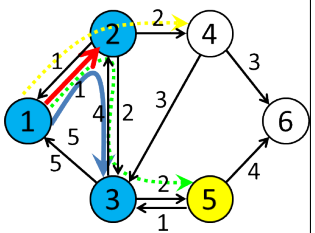
- ステップ (2) では，まず，確定した点 3 を N から S に移します．

ダイクストラ法

3 回目（2）

$S = \{1, 2, 3\}$, $N = \{4, 5, 6\}$

$d(5) = \infty > d(3) + w_{35} = 3 + 2 = 5$
$\rightarrow d(5) \leftarrow 5$, $p(5) \leftarrow 3$



- さて，点 3 ですが，全部で 6 本の枝が出入りしています．多くの枝が出入りしていて，少々ややこしいのですが，点 3 から枝で直接つながっている N の点は，点 5 だけです．点 3 からは点 5 が直接つながっています．
- 点 5 に関しては，現在 $d(5) = \infty$ で，点 3 から点 5 への距離 w_{35} は 2 です．
- $d(3) + w_{35} = 3 + 2 = 5$ は $d(5)$ より短いので，$d(5)$ を 5 として，暫定最短路を経路 1， 2， 3， 5 とします．
- 点 5 までの暫定最短路における点 5 の直前の点を 3 とします．すなわち，$p(5)$ を 3 とします．
-

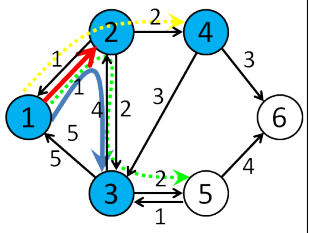
ダイクストラ法

4 回目（1）

$S = \{1, 2, 3\}$, $N = \{4, 5, 6\}$

$d(4) = 3, d(5) = 5$
$d(6) = \infty$

$\min\{d(4), d(5), d(6)\}$
$= \min\{3, 5, \infty\} = 3$
より $v = 4$

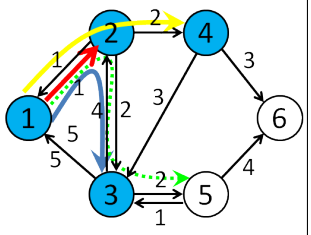


- ここから 4 回目の繰り返しです.
- まず，終了条件は満たしていないので，N の中から d が最短になる点 v を選びます.
- ここでは，$d(4)$ が 3 で最短です．すなわち，v は 4 となります.

ダイクストラ法

4回目（2）

$S \leftarrow \{1,2,3,4\}$, $N \leftarrow \{5,6\}$



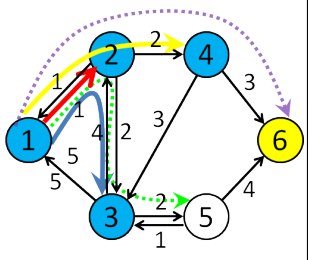
- ステップ (2) では，まず，確定した点 4 を N から S に移します．

ダイクストラ法

4 回目（2）

$S = \{1, 2, 3, 4\}$, $N = \{5, 6\}$

$d(6) = \infty > d(4) + w_{46} = 3 + 3 = 6$
$\rightarrow d(6) \leftarrow 6$, $p(6) \leftarrow 4$



- 次に，点 4 から枝で直接つながっている N の点，点 4 からは点 6 が直接つながっています．
- 点 6 に関しては，現在 $d(6) = \infty$ で，点 4 から点 6 への距離 w_{46} は 3 です．
- $d(4) + w_{46} = 3 + 3 = 6$ は $d(6)$ より短いので，$d(6)$ を 6 として，暫定最短路を経路 1， 2， 4， 6 とします．
- 点 6 までの暫定最短路における点 6 の直前の点は 4 ですから $p(6)$ を 4 とします．

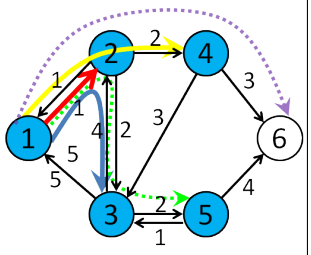
ダイクストラ法

5 回目（1）

$S = \{1, 2, 3, 4\}$, $N = \{5, 6\}$

$d(5) = 5$
$d(6) = 6$

$\min\{d(5), d(6)\} = \min\{5, 6\} = 5$
より $v = 5$

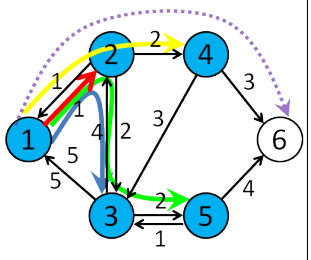


- ここから 5 回目の繰り返しです.
- まず，終了条件は満たしていないので，N の中から d が最短になる点 v を選びます.
- ここでは，$d(5)$ が 5 で最短です．すなわち，v は 5 となります.

ダイクストラ法

5 回目（2）

$S \leftarrow \{1,2,3,4,5\}$, $N \leftarrow \{6\}$



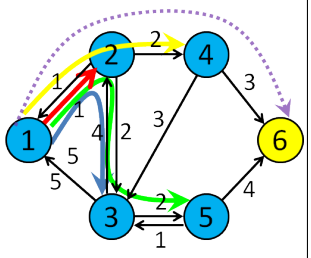
- ステップ (2) では，まず，確定した点 5 を N から S に移します．

ダイクストラ法

5 回目（2）

$S = \{1, 2, 3, 4, 5\}$, $N = \{6\}$

$d(6) = 6 < d(5) + w_{56} = 5 + 4 = 9$
$\rightarrow$ $d(6) = 6$, $p(6) = 4$ のまま



- 次に，点 5 から枝で直接つながっている N の点，点 5 からは点 6 が直接つながっています．
- 点 6 に関しては，現在 $d(6) = 6$ で，点 5 から点 6 への距離 w_{56} は 4 です．
- $d(5) + w_{56} = 5 + 4 = 9$ は $d(6)$ より長いので，暫定最短路はそのまま，$d(6)$ は 6，$p(6)$ は 4 のままです．
-

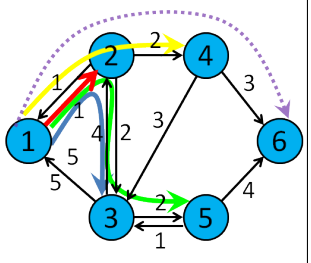
ダイクストラ法

6 回目（1）

$S = \{1, 2, 3, 4, 5\}$, $N = \{6\}$

$d(6) = 6$

$\min\{d(6)\} = \min\{6\} = 6$
より $v = 6$

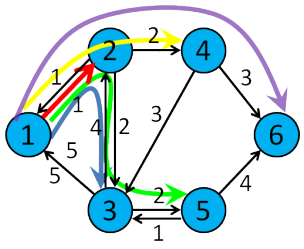


- ここから 6 回目の繰り返しです.
- まず，終了条件は満たしていないので，N の中から d が最短になる点 v を選びますが，もう点 6 しか残っていないので，v は 6 となります.

ダイクストラ法

6 回目（2）

$S \leftarrow \{1,2,3,4,5,6\}$, $N \leftarrow \{\}$



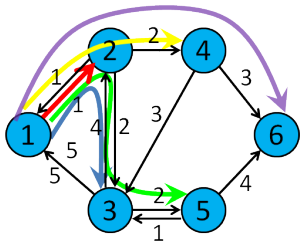
- 確定した点 6 を N から S に移します．
- N には点が残っていないので，このステップは終わりです．

ダイクストラ法

7回目（1）

$S = \{1, 2, 3, 4, 5, 6\}$, $N = \{\}$

$S = V$ なので終了

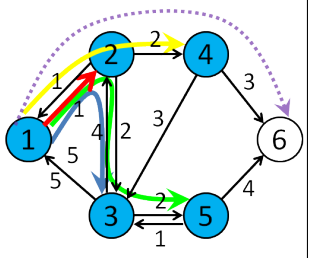


- 7回目の繰り返しですが，$S = V$ となりましたので，終了条件が満たされ，これで終了です．

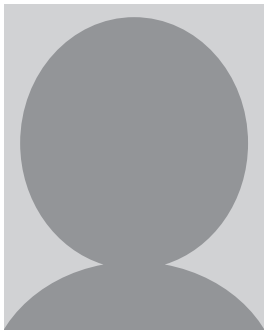
ダイクストラ法

- 点 6 への最短路
- $d(6) = 6$ なので距離は 6
- $p(6) = 4$, $p(4) = 2$,
  $p(2) = 1$ なので，

$$6 \leftarrow 4 \leftarrow 2 \leftarrow 1$$
$$\Downarrow$$
$$1 \rightarrow 2 \rightarrow 4 \rightarrow 6$$

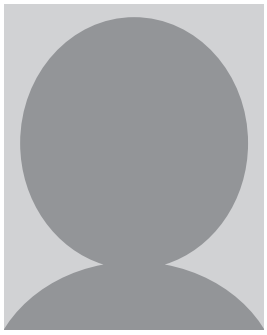


- 例えば，点 6 への最短路は $d(6) = 6$ から距離は 6 です．
- 経路は点 6 から始点へ最短路を逆にたどると，$p(6) = 4$，
  $p(4) = 2$，$p(2) = 1$ ですから，6, 4, 2, 1，
  始点からなら，1, 2, 4, 6 となります．



- 以上，最短路問題を効率的に解くダイクストラ法について紹介しました．
- 始点に近い点から順次最短路が確定されていく様（さま）が見てとれたと思います．

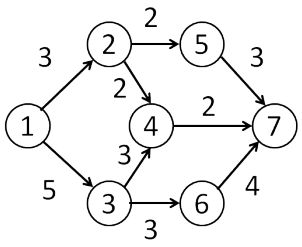
最大流問題



- 今度は，これもまたネットワーク最適化法の代表的な例題である最大流問題について説明します.

最大流問題

- 点は地点，枝は通路，重みは通路の容量（流量の最大値，単位時間に通れる人数）
- 地点 1 に客が集まっていて，出口は地点 7
- 最も多く出口に客を送り出すよう客を誘導

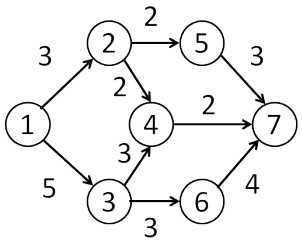


- 最大流問題とは次のような問題です．
- 図において，点は地点，枝は通路，重みは容量，ここではその通路を単位時間に通れる人数の上限を表します．
- 今，地点 1 に客が集まっていて，地点 7 が出口とします．
- 最も多く出口に客を送り出すにはどのように客を誘導するか，つまりどの経路にどれだけの客を通すかという問題です．
- ここでは，非常時における客の避難誘導のようなことを想定しています．
- 他にも，できるだけ多くの自動車を渋滞させないように，複数の経路に振り分ける，
- 通信ネットワークで，帯域の狭い複数の経路を用いてできるだけ高速な通信を実現するなど，最大流問題は様々な分野に存在します．
- それでは，最大流問題を定式化していきます．

最大流問題

- 点の集合を V，枝の集合を E，
- 枝 (i,j) の枝 (i,j) の流量を x_{ij}
  容量を u_{ij}
- 流量 f は，始点 s および終点 t の流量に等しい

$$\sum_{(s,j)\in E} x_{sj} - \sum_{(j,s)\in E} x_{js} = f$$
$$\sum_{(t,j)\in E} x_{tj} - \sum_{(j,t)\in E} x_{jt} = -f$$



- まず，決定変数を定義します．
- 枝 (i,j) の流量，この例では通路を通る単位時間当たりの人数を x_{ij} とします．
- 枝 (i,j) の容量，すなわち流量の最大値を u_{ij} とします．
- このネットワークにおける始点から終点までの流量 f は，始点 s における流出量および終点 t における流入量と等しくなります．
- このグラフは有向グラフで，始点である点 1 からは，点 2 と点 3 へ出る枝がありますが，点 1 に入る枝はありません．
- つまり，この図の例においては，最初から流入はあり得ないのですが，一般的には始点に入る枝も存在します．
- 始点 s における流出量は，s から他の点 j に枝 (s,j) を通して出て行く量 x_{sj} の合計から，他の点 j から s に枝 (j,s) を通して入ってくる量 x_{js} の合計を引いたものになります．
- したがって，x_{sj} の j に関する和マイナス x_{js} の j に関する和イコール f となります．
- 終点 t に関しても同様で，他の点 j から t に枝 (j,t) を通して入ってくる量 x_{jt} の合計から t から他の点 j に枝 (t,j) を通して出て行く量 x_{tj} の合計，を引いたものになります．
- したがって，始点の書き方に合わせて，流出から書くようにしますと，x_{tj} の j に関する和マイナス x_{jt} の j に関する和イコール $-f$ となります．
-

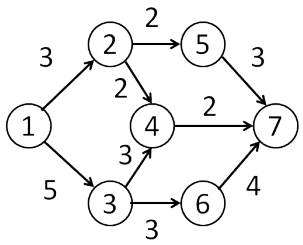
最大流問題

- 他の点 $i \in V \backslash \{s,t\}$ においては，点に入る流量と，点から出る流量が等しい

$$\sum_{(i,j)\in E} x_{ij} - \sum_{(j,i)\in E} x_{ji} = 0$$

- 各枝の流量 x_{ij} は容量 u_{ij} 以下

$$x_{ij} \leq u_{ij} \quad (i,j) \in E$$



- 次は，始点と終点以外の点に関する制約です．
- 始点 s と終点 t 以外の点 i では，他の点から入ってくる量と他の点に出て行く量が等しくなります．
- すなわち，i から他の点 j に枝 (i,j) を通して出て行く量 x_{ij} の合計と，他の点 j から i に枝 (j,i) を通して入ってくる量 x_{ji} の合計は等しくなります．
- したがって，x_{ij} の j に関する和マイナス x_{ji} の j に関する和イコール 0 となります．
- それから，各枝の流量は容量以下でなくてはなりません．すなわち，すべての枝 (i,j) に関して $x_{ij} \leq u_{ij}$ となります．
- また，ここに書いてありませんが，流量 x_{ij} は非負です．
- これらをまとめると，最大流問題は次のように定式化されます．
-

最大流問題

最大化	f	流量
制約条件	$\sum_{(s,j)\in E} x_{sj} - \sum_{(j,s)\in E} x_{js} = f$	始点 s
	$\sum_{(t,j)\in E} x_{tj} - \sum_{(j,t)\in E} x_{jt} = -f$	終点 t
	$\sum_{(i,j)\in E} x_{ij} - \sum_{(j,i)\in E} x_{ji} = 0$	$i \in V\backslash\{s,t\}$
	$0 \le x_{ij} \le u_{ij}$ for $(i,j) \in E$	容量制約

- 目的関数は流量 f で，f を最大化する各枝の流量を求めます.
- 決定変数は，x_{ij} です.
- 制約条件は，上から，始点 s，終点 t，その他の点 i の流量に関する条件, 最後が容量の制約と非負条件です.
- 実はこの定式化は冗長で，もっとコンパクトにまとめることができます.

最大流問題

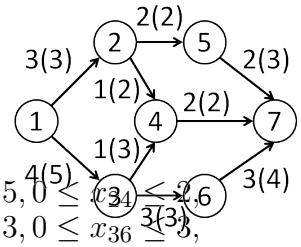
最大化	$f = \sum_{(s,j)\in E} x_{sj}$	流量
制約条件	$\sum_{(i,j)\in E} x_{ij} - \sum_{(j,i)\in E} x_{ji} = 0$	$i \in V\backslash\{s,t\}$
	$0 \le x_{ij} \le u_{ij}$ for $(i,j) \in E$	容量制約

- 最大流問題はこのようにコンパクトに定式化されます.
- まず，目的関数の流量は始点からの流出量となっていますが，x_{sj} のみの和となっており，x_{js} の和が消えています.
- この問題において，始点に他の点から流入することはおかしいので，当然と言えば当然です.
-
- それから，終点に関する制約が消えていますが，他の点に関する制約が満たされていれば，終点に関する制約は必然的に満たされるので，敢えて加える必要がないからです.
- これでかなりすっきりとしました.
- これは線形最適化問題になっています.
- すなわち，最大流問題は線形最適化問題として定式化することができます.
- 最大流問題の一般的な定式化ができましたので，今度はより具体的に定式化します.

最大流問題

最大化 $f = x_{12} + x_{13}$

制約条件
$(x_{24} + x_{25}) - x_{12} = 0$
$(x_{34} + x_{36}) - x_{13} = 0$
$x_{47} - (x_{24} + x_{34}) = 0$
$x_{57} - x_{25} = 0$
$x_{67} - x_{36} = 0$
$0 \leq x_{12} \leq 3, 0 \leq x_{13} \leq 5, 0 \leq x_{24} \leq 2,$
$0 \leq x_{25} \leq 2, 0 \leq x_{34} \leq 3, 0 \leq x_{36} \leq 3,$
$0 \leq x_{47} \leq 2, 0 \leq x_{57} \leq 3, 0 \leq x_{67} \leq 4.$



- 図の枝についている数字のうち，括弧に囲まれている数値はその枝の容量を表しています．括弧の外の数値は，ネットワークの流量が最大になる時のその枝の流量，すなわち最適解を表しています．
- まず，目的関数 f は始点である点 1 からの流出量です．点 1 は点 2 および点 3 と枝で繋がっていますから，点 1 からの流出量 f は，

$$f = x_{12} + x_{13}$$

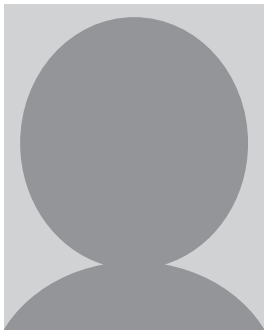
  です．
- 次に始点でも終点でもない点，すなわち点 2 から点 6 までの制約条件を定式化します．
- まず，点 2 は点 1 から流入し，点 4，点 5 に流出しています．
- 点 2 における流入と流出の量は等しくなります．
- そのため点 2 における流入と流出の関係は，

$$(x_{24} + x_{25}) - x_{12} = 0$$

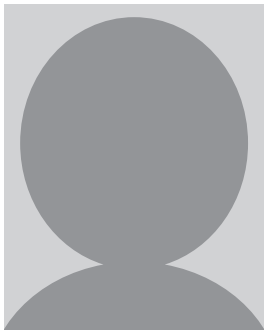
  となります．
- 点 3, 4, 5, 6 に関しても同様です．
- 容量の制約と非負条件もすべての枝について記述します．
- 以上で，この最大流問題を線形最適化問題として定式化することができました．
- 次，まとめで顔出し

20 秒版

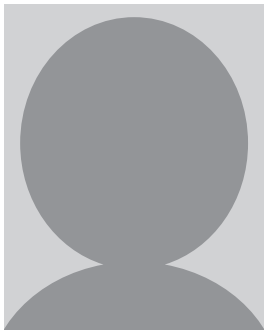
- 今回はネットワーク最適化法についてお話しました.
- 今回紹介できたのはネットワーク最適化法のほんの一部にすぎません.
- 印刷教材の演習問題に最小費用流問題を載せてありますので，ご参照ください.
- 今回はこれで終わります.



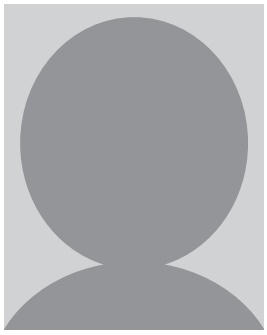
- 今回の講義では，ネットワーク最適化法の初歩として，グラフによるネットワークの表現，最短路問題の整数最適化問題および線形最適化問題としての定式化，最短路問題を効率的に解くダイクストラ法，最大流問題の線形最適化問題としての定式化についてお話しました．



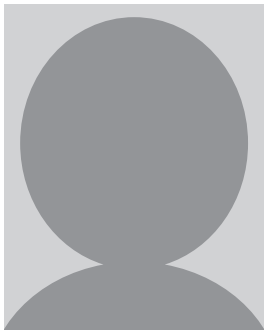
- 今回は最短路問題と最大流問題だけしか紹介できませんでしたが，他にもネットワーク最適化問題はあります．
- 一部は，組み合わせ最適化法の回に取り上げます．また，印刷教材の演習問題に最小費用流問題を載せてありますので，ぜひ解答とともにご覧下さい．



- また，問題の構造を利用した効率的な解法は，最短路問題におけるダイクストラ法だけ詳しく説明しましたが，最短路問題の効率的な解法はダイクストラ法以外にも幾つか存在します．
- また，最大流問題にも印刷教材で取り上げているフロー増加法をはじめ，効率的な解法が幾つかあります．
- ネットワーク最適化問題の効率的な解法は盛んに研究されていますので，興味のある方は参考文献をご覧ください．



- 現実の問題解決では，問題を定式化することから始まりますが，
- その際，これは最短路問題ですとか，最大流問題ですとはもちろん書かれていません．
- まず，きっちりと数理最適化問題として定式化して，その上で，最短路問題や最大流問題といった効率的な解法が存在する問題に当てはまるか調べるという手順をとることになるでしょう．
- これはネットワーク最適化法に限った事ではありませんが，最初から最短路問題と知っていて定式化するのとは異なり，決定変数をどう定義するか，制約をどのように数式で表現するかなどを自分で考える必要があります．
- その際，今回紹介した最短路問題と最大流問題や，組み合わせ最適化問題として取り上げる問題の定式化は参考になると思います．



- ダイクストラ法などのプログラムも探せば見つかるかもしれませんが，線形最適化法や整数最適化法の汎用的なソルバーを用いても問題を解くことはできます．ソルバーに関しては第 3 回の話具合で加える
- ソルバーを用いて，例題や自分で定式化した問題を解いてみることは，ネットワーク最適化法の理解を深めるのに有効ですので，ぜひ実行してみて下さい．
- 終了