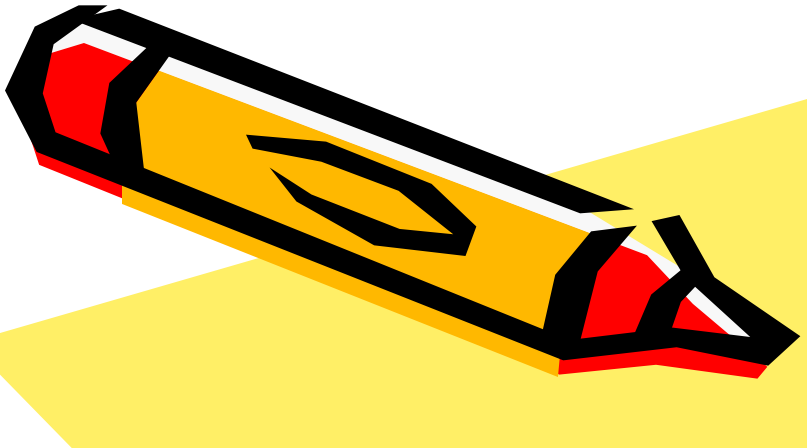


数理計画法E（第6学期）
第6回

担当：飯田勝吉（いいだかつよし）
iida@gsic.titech.ac.jp

課題の回答

- 取引先Biからの注文料をΔ増やした場合、新しい問題の目的関数の最小値$(y^*)'_i$ は$\bar{z}' = \bar{z} + \Delta(y^*)'_i$ となる。ただし、$\bar{z}'$ は双対問題の最適解 y^*の要素である。よって、コスト増を最小に抑えるためには $\min_i (y^*)'_i$ を与えるBiへの注文料を増やせばよい。
- 実際に計算すると、$y^*=(0,3,0,4,3)$となるため、$(y^*)'_1, (y^*)'_3$ が最小値0をとる。よって、取引先1または3が解。

標準形(1)

- [問題 2.1] 2つの工場 A1, A2で同じ製品を生産し、3つの取引先B1,B2,B3へ納入している会社がある。各取引先からの注文量、各工場における生産量、および各工場から各製品までの輸送コストは表のとおりである。層輸送コストを最小とする輸送計画とは？

(a) 注文量

B1	70
B2	40
B3	60

(b) 生産量

A1	90
A2	80

(c) 輸送コスト

	B1	B2	B3
A1	4	7	12
A2	11	6	3

標準形(2)

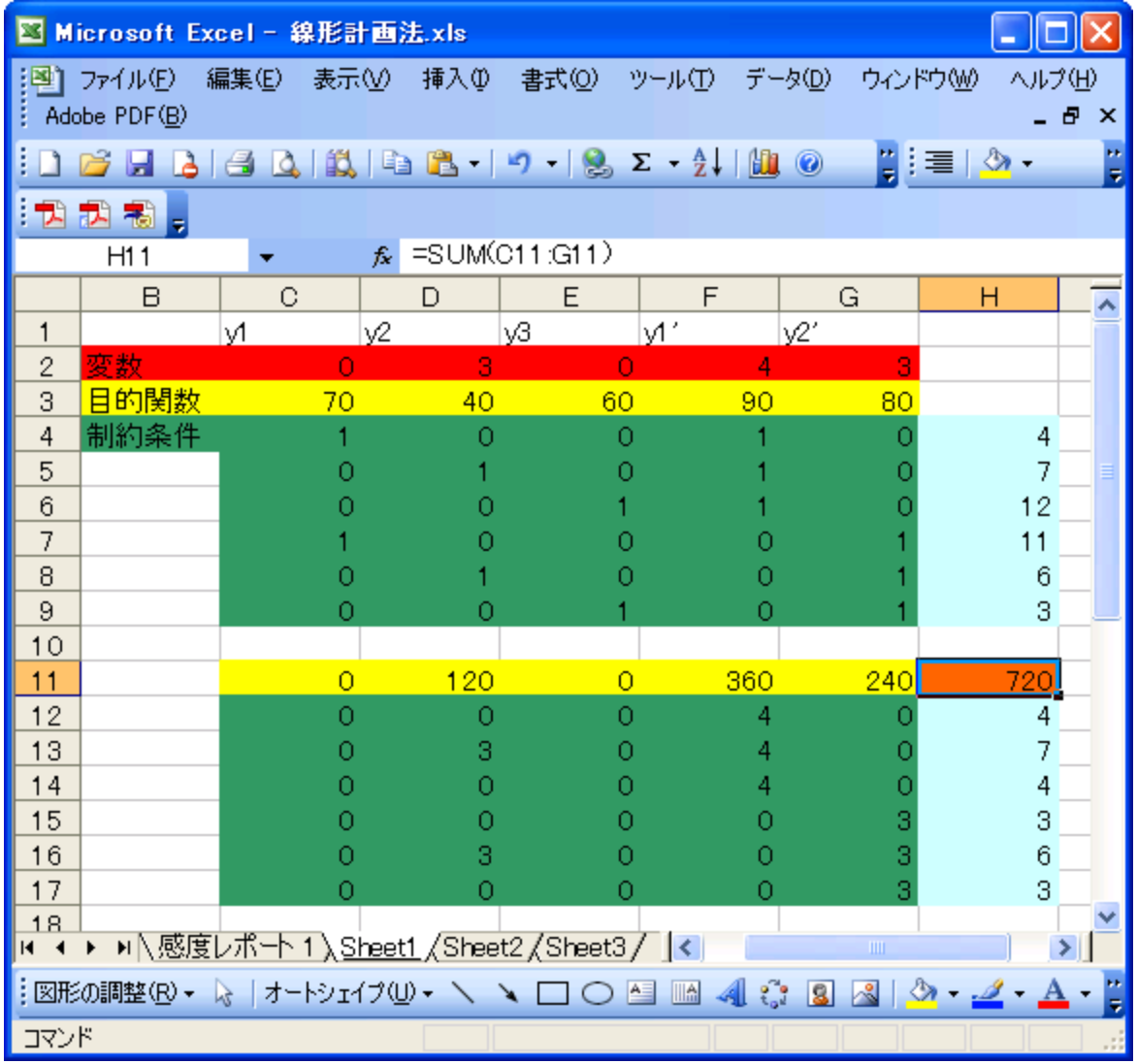
- 工場Aiから取引先Bjへ輸送する量を x_{ij}（単位）とすると、この問題は

目的関数: $4x_{11}+7x_{12}+12x_{13}+11x_{21}+6x_{22}+3x_{23} \rightarrow$ 最小化

制約条件: $x_{11}+x_{21}=70$　　$x_{11}+x_{12}+x_{13}=90$

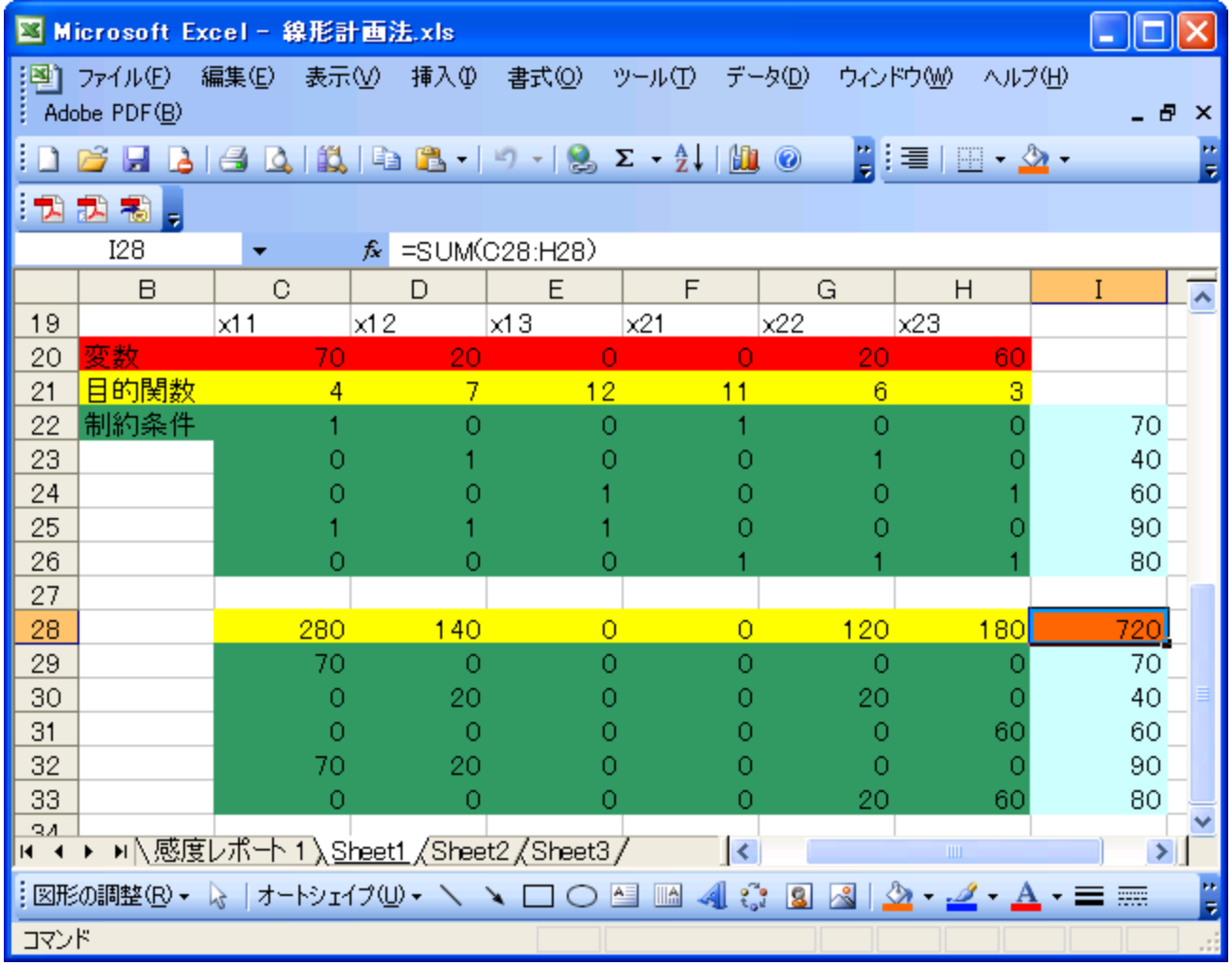
$x_{12}+x_{22}=40$　　$x_{21}+x_{22}+x_{23}=80$

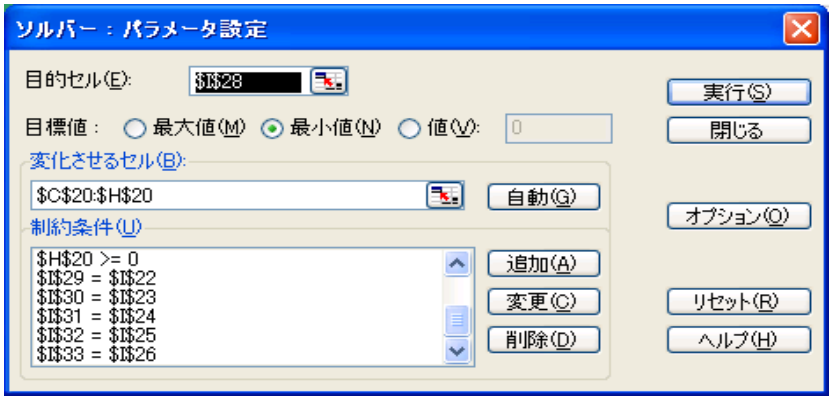
$x_{13}+x_{23}=60$　　$x_{ij} \geq 0 \quad (i=1,2;\ j=1,2,3)$



	B	C	D	E	F	G	H
1		y1	y2	y3	y1'	y2'	
2	変数	0	3	0	4	3	
3	目的関数	70	40	60	90	80	
4	制約条件	1	0	0	1	0	4
5		0	1	0	1	0	7
6		0	0	1	1	0	12
7		1	0	0	0	1	11
8		0	1	0	0	1	6
9		0	0	1	0	1	3
10							
11		0	120	0	360	240	720
12		0	0	0	4	0	4
13		0	3	0	4	0	7
14		0	0	0	4	0	4
15		0	0	0	0	3	3
16		0	3	0	0	3	6
17		0	0	0	0	3	3

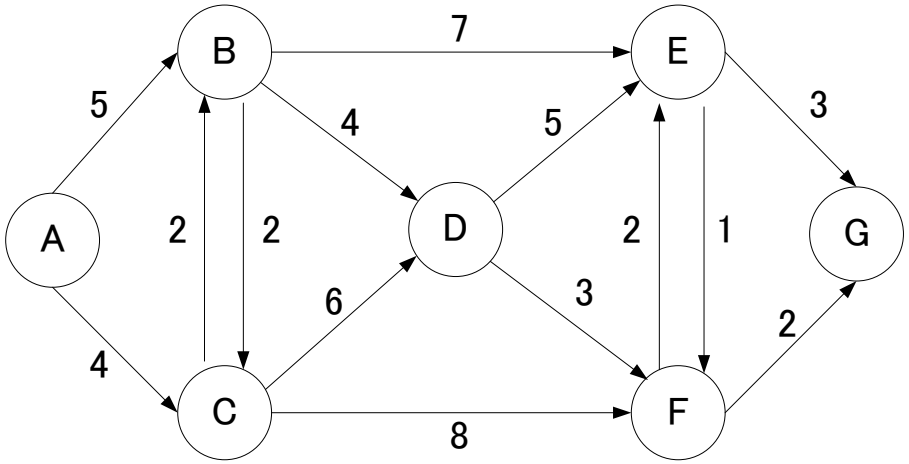
- エクセルの「ソルバー」の機能を利用


Microsoft Excel - 線形計画法.xls
ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)
Adobe PDF(B)
I28
=SUM(C28:H28)
B C D E F G H I
19 x11 x12 x13 x21 x22 x23
20 変数 70 20 0 0 20 60
21 目的関数 4 7 12 11 6 3
22 制約条件 1 0 0 1 0 0 70
23 0 1 0 0 1 0 40
24 0 0 1 0 0 1 60
25 1 1 1 0 0 0 90
26 0 0 0 1 1 1 80
27
28 280 140 0 0 120 180 720
29 70 0 0 0 0 0 70
30 0 20 0 0 20 0 40
31 0 0 0 0 0 60 60
32 70 20 0 0 0 0 90
33 0 0 0 0 20 60 80
感度レポート 1 Sheet1 Sheet2 Sheet3
図形の調整(R) オートシェイプ(U)
コマンド


ソルバー：パラメータ設定
目的セル(E): $I$28
目標値： 最大値(M) 最小値(N) 値(V): 0
変化させるセル(B):
$C$20:$H$20
自動(G)
制約条件(U):
$H$20 >= 0
$I$29 = $I$22
$I$30 = $I$23
$I$31 = $I$24
$I$32 = $I$25
$I$33 = $I$26
追加(A)
変更(C)
削除(D)
実行(S)
閉じる
オプション(O)
リセット(R)
ヘルプ(H)

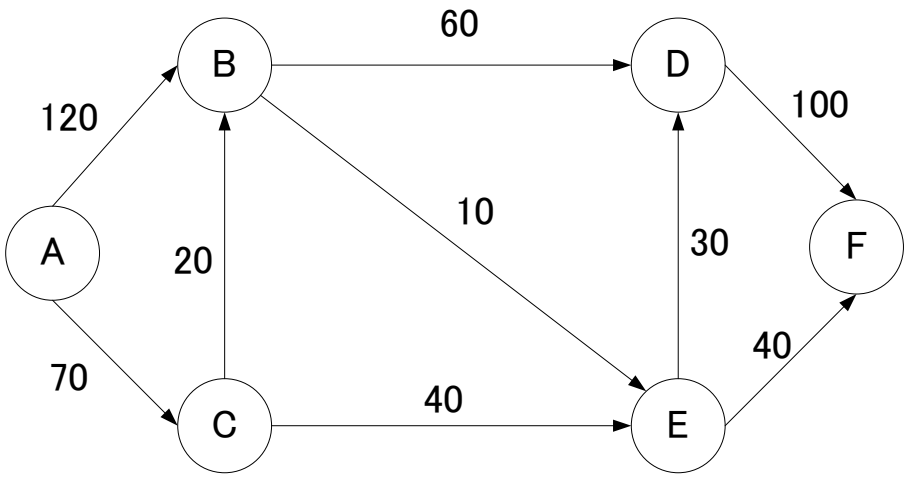
ネットワーク計画法

- 最短路問題
  - 節点Aから節点G間で最短で行くための経路を求めよ。ただし、枝に与えられた値はその枝の長さを表す。



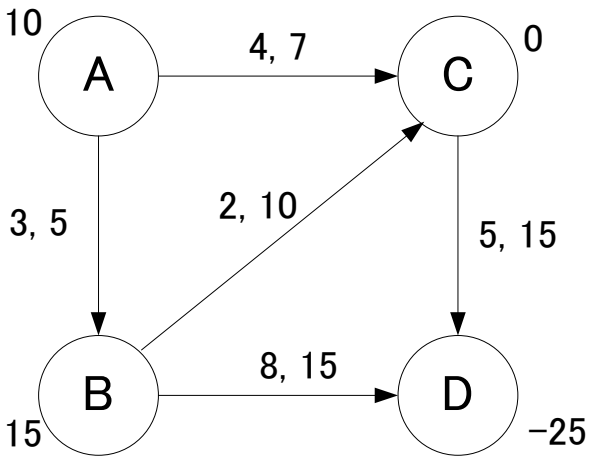
ネットワーク計画法

- 最大流問題
  - 節点Aから節点Fまで最大どれだけの流量を流すことが出来るか。ただし、枝に与えられた値はその枝の容量を示す。



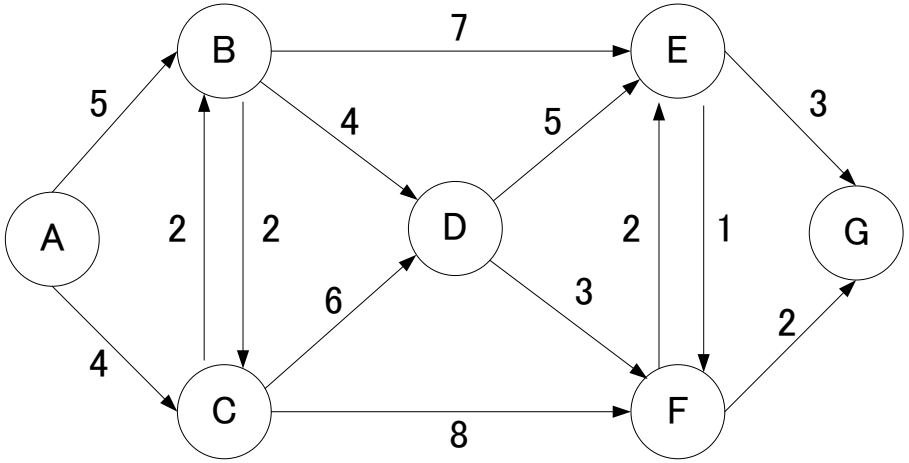
ネットワーク計画問題

- 最小費用流問題
  - 全ての節点における需要量・供給量を満足しつつコストを最小にするにはどうしたらよいか。(枝の値＝(コスト、容量)、節点に与えられた値＝供給量(正)、需要量(負))



最短路問題(1)

- [問題3.1]節点Aから節点E間で最短で行くための経路を求めよ。ただし、枝に与えられた値はその枝の長さを表す。



最短路問題(2)
定式化

- 与えられたグラフG=(V,E)において、接点の数|V|=n, 枝の数|E|=mとする。枝(I,j)の長さをaijとし、経路の

始まりの点＿＿＿＿＿＿をs∈V

終わりの点＿＿＿＿＿＿をt∈V

定式化

目的関数：
$$\sum_{(i,j)\in E} a_{i,j}x_{i,j} \rightarrow \text{最小化}$$

制約条件：
$$\sum_{(s,j)\in E} x_{s,j} - \sum_{(i,s)\in E} x_{i,s} = 1$$
$$\sum_{(v,j)\in E} x_{v,j} - \sum_{(i,v)\in E} x_{i,v} = 0 \quad (v \neq s,t)$$
$$\sum_{(t,j)\in E} x_{t,j} - \sum_{(i,t)\in E} x_{i,t} = -1$$
$$\mathbf{x} \geq \mathbf{0}$$

最短路問題(3)

- Xij
  - 経路が枝i,jを包含　　→1
  - 経路が枝i,jを包含せず→0
- 目的関数
  - 経路に含まれる＿＿＿＿＿＿＿＿＿＿を最小とする
- 制約条件
  - 各制約条件の一つ目の和＝
    - 注目している節点から＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿
  - 各制約条件の二つ目の和＝
    - 注目している節点に＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿
  - 始点では、その差が1、終点では−1、その他の節点では0

これより、線形計画問題としてシンプレックス法を用いて解くことが可能

最短路問題(4)
ダイクストラのアルゴリズム

- Dijkstra's algorithm
  - 枝に長さが与えられたグラフG=(V,E)と一つの始点 $s \in V$ が与えられたとき、始点から任意の節点への最短路を求めるアルゴリズム
  - 木 (tree)
    - 閉路(ループ)を持たないグラフ。特にあるグラフ G=(V,E) に対して全ての節点を含む木をスパニングツリー(spanning tree)という
  - 最短路木
    - 一つの接点から各接点への最短路からなるスパニングツリー

EthernetにおけるSpanning Tree Protocol (STP): IEEE 802.1d（ループ解決）

最短路問題(5)
ダイクストラのアルゴリズム

- 始点から順に隣接する節点にラベルをつけながら最短路木をグラフ全体に張る

- (Step 0)
  - 初期状態を設定する
  - $S=\{\}$, $d(s)=0$, $d(i)=\infty$ for all $i \in V \setminus \{s\}$
- (Step 1)
  - $S=V$ならば終了。そうでないならばSを除外したVから$d(i)$が最小の i を選択し、$\hat{i}$ とする。つまり

$$d(\hat{i}) = \min(d(i) \mid i \in V \setminus S)$$

- (Step 2)
  - $S \leftarrow S \cup \{\hat{i}\}$とする。$(\hat{i}, j) \in E$ である全ての $j \notin S$ に対して

$$d(j) > d(\hat{i}) + a_{\hat{i},j} \text{ならば} d(j) \leftarrow d(\hat{i}) + a_{\hat{i},j},\ p(j) \leftarrow \hat{i}$$

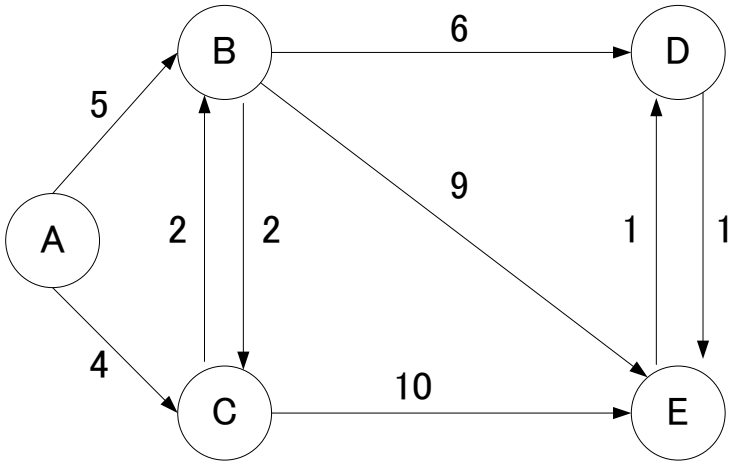
  とする。
- (Step 3)
  - Step 1に戻る。

最短路木(6)
ダイクストラのアルゴリズム

- Sは＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿節点の集合
- 節点jに対してd(j)は現在の繰り返しにおける＿＿＿＿＿＿＿＿＿＿
- p(j)は現在計算中の最短路木における節点jへの始点側の隣接節点

最短路木(7)
ダイクストラのアルゴリズム

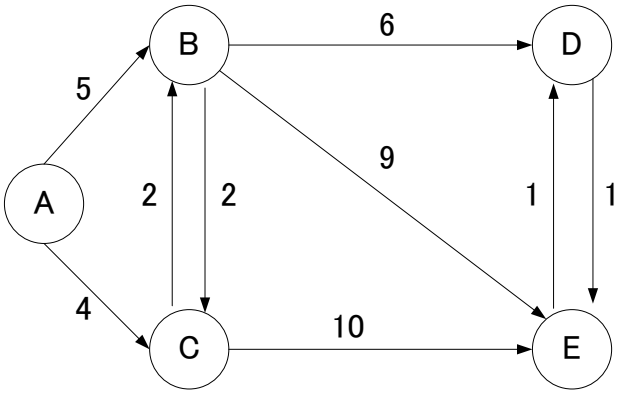
- 例:V{A,B,C,D,E}, 始点＝A



(0) 初期状態: S={}, d(A)=0,
d(B)=d(C)=d(D)=d(E)=∞

(0,∞,∞,∞,∞)

最短路木(8)
ダイクストラのアルゴリズム

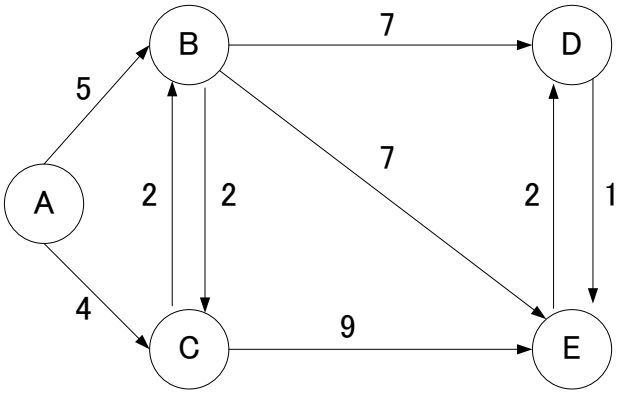


現在の最短路木

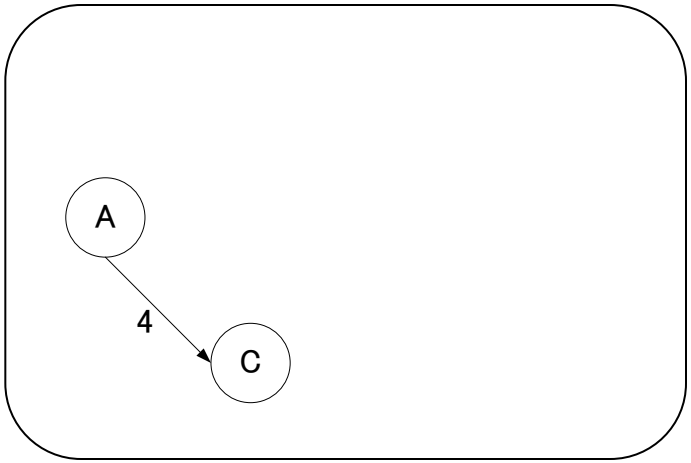
(1) $\min(0,\infty,\infty,\infty,\infty)=0$より、$\hat{i}=A$を選択し、
S={A}。(A,j)∈Eであるjはと
d(B)>d(A)+5より、d(B)=5, p(B) = A
d(C)>d(A)+4より、d(C)=_____,p(C)=_____

(0,5(A),4(A),∞,∞)

最短路木(9)
ダイクストラのアルゴリズム



現在の最短路木



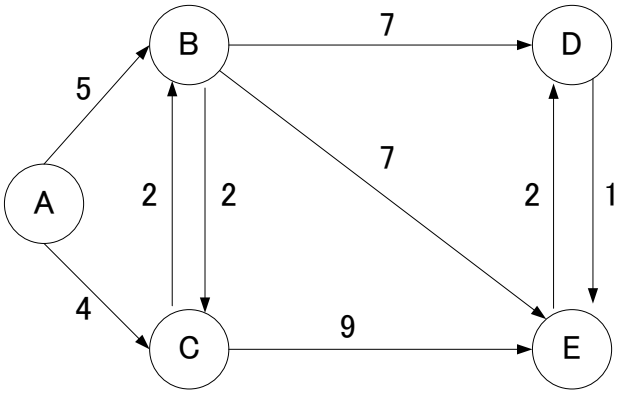
(2) $\min(5, 4, \infty, \infty, \infty)=4$より、 $\hat{i}=$を選択し、
$S=\{A, C\}$。$(C,j)\in E$であるjはBとE

d(B)＿＿d(C)+2より、d(B)=＿＿, p(B) =＿＿

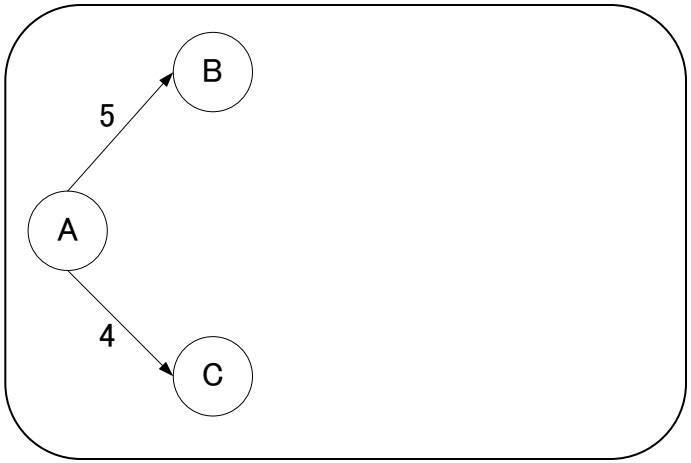
d(E)＿＿d(C)+9より、d(E)=＿＿,p(E)=＿＿

(0,5(A),4(A),∞,13(C))

最短路木（10）
ダイクストラのアルゴリズム



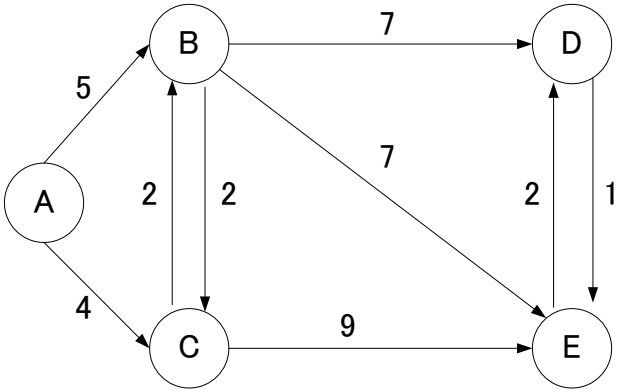
現在の最短路木



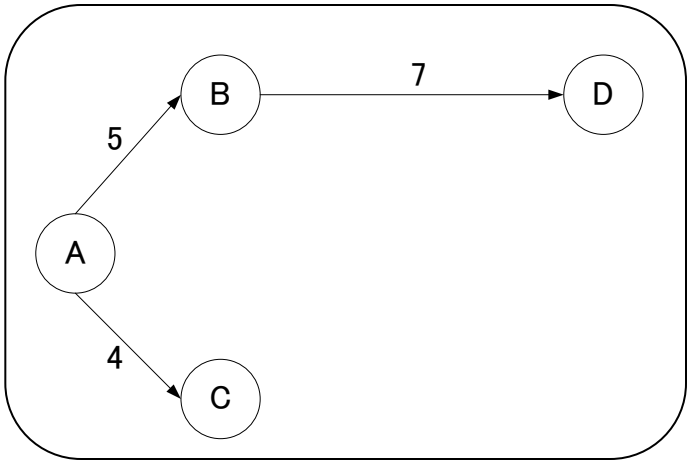
(3) $\min(5, \infty, 13)=5$より、$\hat{i}=$____を選択し、
S={_____}。(____,j)∈Eであるjはと
d(D)____d(B)+7より、d(D)=____, p(D) =____
d(E)____d(B)+7より、d(E)=____,p(E)=____

(0,5(A),4(A),12(B),12(B))

最短路木(11)
ダイクストラのアルゴリズム



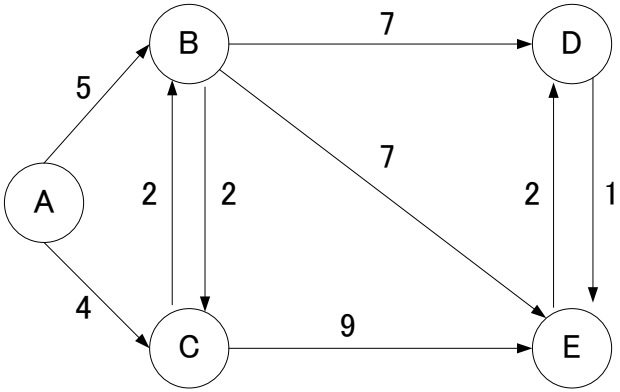
現在の最短路木



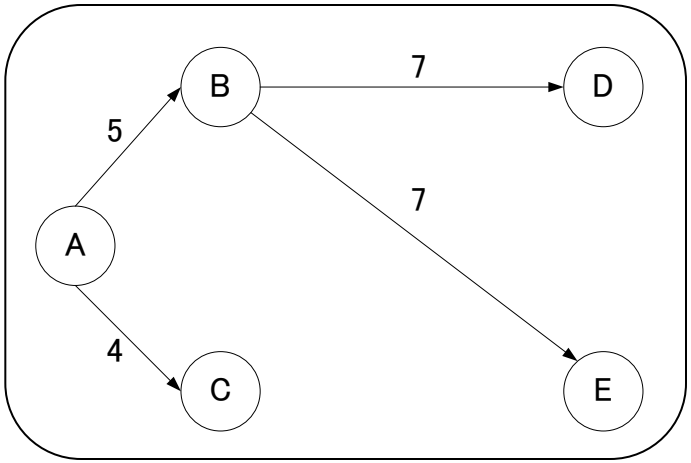
(4) min(12,12)=12より、 $\hat{i}$ = ____または____。ここでは、Dを選択すると、S={_____}
d(E)____d(D)+1より、d(E)=____, p(E) =____

(0,5(A),4(A),12(B),12(B))

最短路木(12)
ダイクストラのアルゴリズム



現在の最短路木

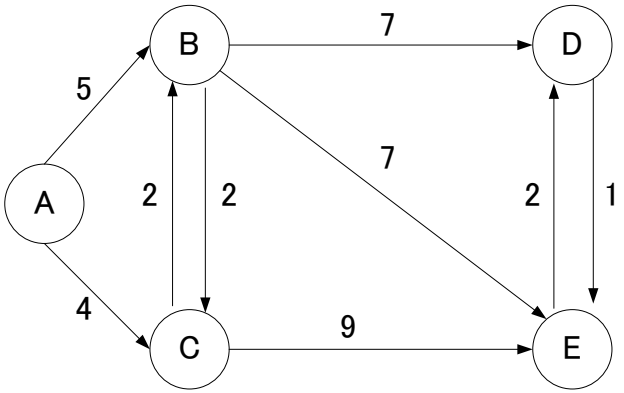


(5) $\hat{i}$ = ＿＿。よってS={＿＿＿＿＿＿＿＿}。S=Vなので終了。

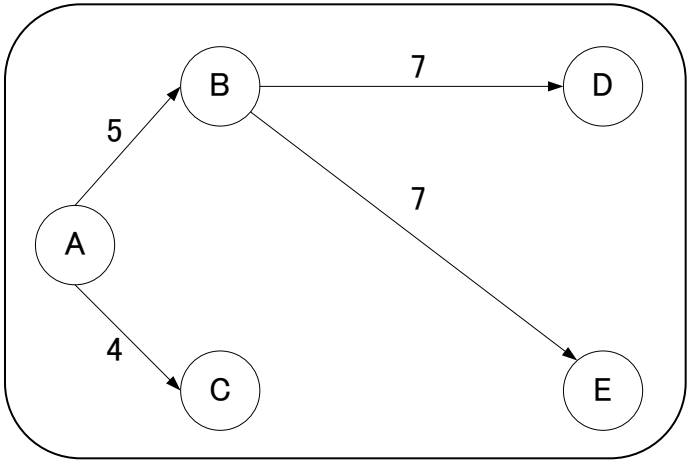
(0,5(A),4(A),12(B),12(B))

最短路木(13)
ダイクストラのアルゴリズム

	A	B	C	D	E
(0)	0	∞	∞	∞	∞
(1)		5(A)	4(A)	∞	∞
(2)		5(A)		∞	13(C)
(3)				12(B)	12(B)
(4)					12(B)
(5)					



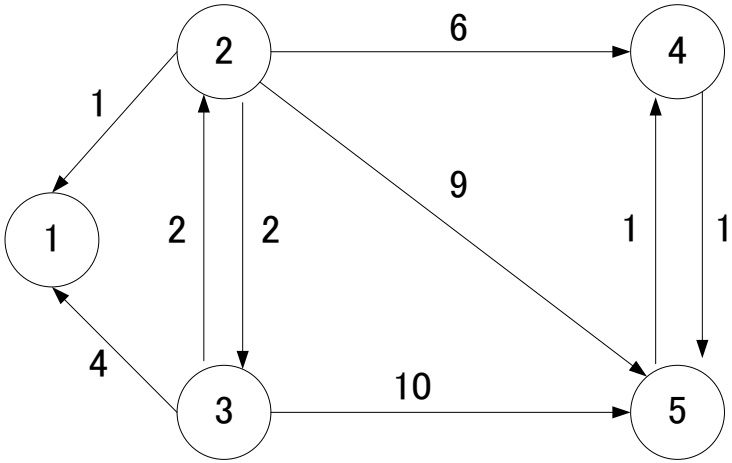
最短路木



下線は次の繰り返しによって
選ばれた節点と始点からの最短距離

課題

- 下図において節点3を始点とする最短路木を求めよ。



ネットワーク経路制御とダイクストラアルゴリズム（１）

- 第２層データリンク層（イーサネット）
  - ループの解決
  - IEEE 802.1d STP
- 第３層インターネット層
  - インターネットのEnd-to-endでの経路の決定
  - 静的経路制御と動的経路制御

ネットワーク経路制御とダイクストラアルゴリズム（2）

- 動的経路制御手法
  - 分散している各ノードがネットワーク全体の状態を知るのが困難
  - 経路の安定化が課題
  - 集中制御型、始点経路制御型、Hop-by-Hop型
- Hop-by-Hop型
  - 距離ベクトル方式(Bellman-Fordアルゴリズム）
    - ループ解決が難しい
    - Routing Information Protocol (RIP)
  - リンク状態方式（Link-stateアルゴリズム）
    - ネットワーク内の全ノードが同時にダイクストラアルゴリズムを実行
    - Open Shortest Path First (OSPF)など
  - パスベクトル方式
    - 上記2方式の折衷方式
    - インターネット全体の経路制御に利用（Border Gateway Protocol: BGP)

traceroute (tracert)コマンドと netstat –nr コマンド

ダイクストラアルゴリズムの正当性と計算量(1)

- ダイクストラのアルゴリズムでは、各繰り返しの終了時において
  - d(i)=[Sに含まれる節点のみを経由した場合のsからiへの最短距離] (3.1)
  - となる。また節点iがSに含まれる場合
  - d(i)=[＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿] (3.2)
  - となる。

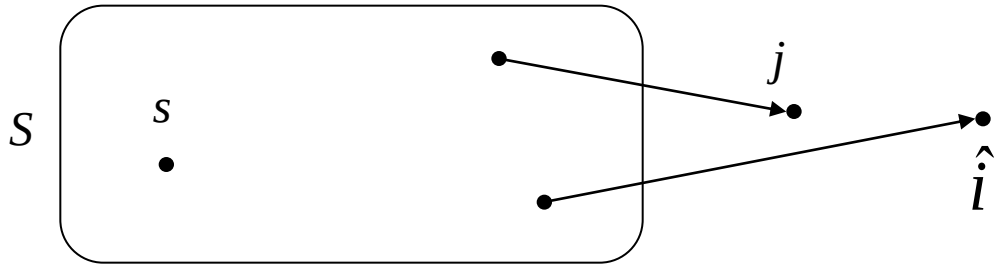
- これを数学的帰納法によって証明する。

ダイクストラアルゴリズムの正当性と計算量(2)

- 1回目の繰り返し終了時
  - S={s}なので、(3.1), (3.2)は共に真。
- k回目の繰り返し開始時
  - (3.1), (3.2)が共に真と仮定。
  - k回目の繰り返しによって選ばれた節点＝$\hat{i}$
  - k回目の繰り返し終了時にどうなるか検討
  - 仮定より$d(\hat{i})$は(3.1)を満たす。
  - もし、$d(\hat{i})$が「真の $\hat{i}$ までの最短距離」ではないとすると、Sに含まれない節点を経由して$\hat{i}$に到達する経路が存在する。(これをrとする)

ダイクストラアルゴリズムの正当性と計算量(3)

- r=(s→S内の節点→S外の節点j→ $\hat{i}$)
- と書けるので、$d(j) \_\_\_ d(\hat{i})$ となるはずである。



- これはこの繰り返しで $\hat{i}$ が選ばれたことに矛盾する。よって、(3.2)式が成立。Q.E.D.

- 以上より、S=Vとなった時点でsから全ての節点に対する最短距離が求まったといえる。

ダイクストラアルゴリズムの正当性と計算量(4)

- 時間計算量と空間計算量
  - 空間計算量(必要メモリー量)
  - 時間計算量(CPU実行時間)
    - _____の回数
- オーダー記法
  - 計算量において支配的な項をとりだしO(f(・))と書く。
  - あるアルゴリズムのステップ数が$T(x)=5x^3+x+\log x$の場合、$O(x^3)$

ダイクストラアルゴリズムの正当性と計算量(5)

- 計算量によるアルゴリズムの分類
  - _____時間アルゴリズム(P問題)
    - ある定数dを使って$O(x^d)$と書ける場合
  - _____時間アルゴリズム(NP問題)
    - $O(2^x)$や$O(x!)$のように問題の大きさxのべき乗の演算が必要な場合
  - NP問題は、問題の大きさxが少し大きくなる(例:2倍、10倍、数十倍)だけで、手に負えなくなる

ダイクストラアルゴリズムの正当性と計算量（6）

- ダイクストラのアルゴリズム
  - 入力パラメータの大きさ|V|=n, |E|=mによって計算量が決定
  - (Step 0)は2n回の代入
  - (Step 1)は繰り返し1回あたり最大n回の比較
  - (Step 2)は全ての繰り返し全体で、m回の比較と高々2m回の代入
  - 繰り返し回数＝n
  - $O(2n+n^2+3m)$
  - $m<n^2$なので、$O(n^2)$
  - つまりこれは、＿＿＿＿＿＿＿＿アルゴリズム

単一のリンクコストを最適化する問題＝P問題
二つ以上のリンクコストを最適化する問題＝NP問題