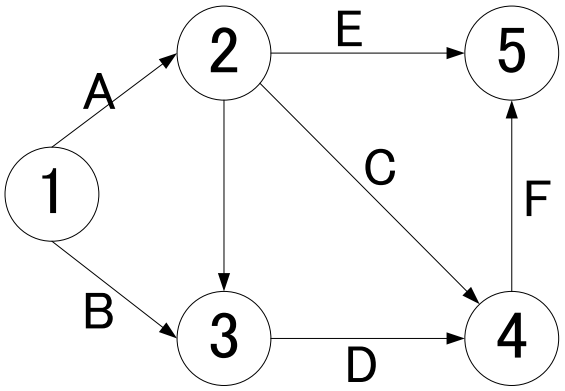
数理計画法E（第6学期）
第10回

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

課題1の回答

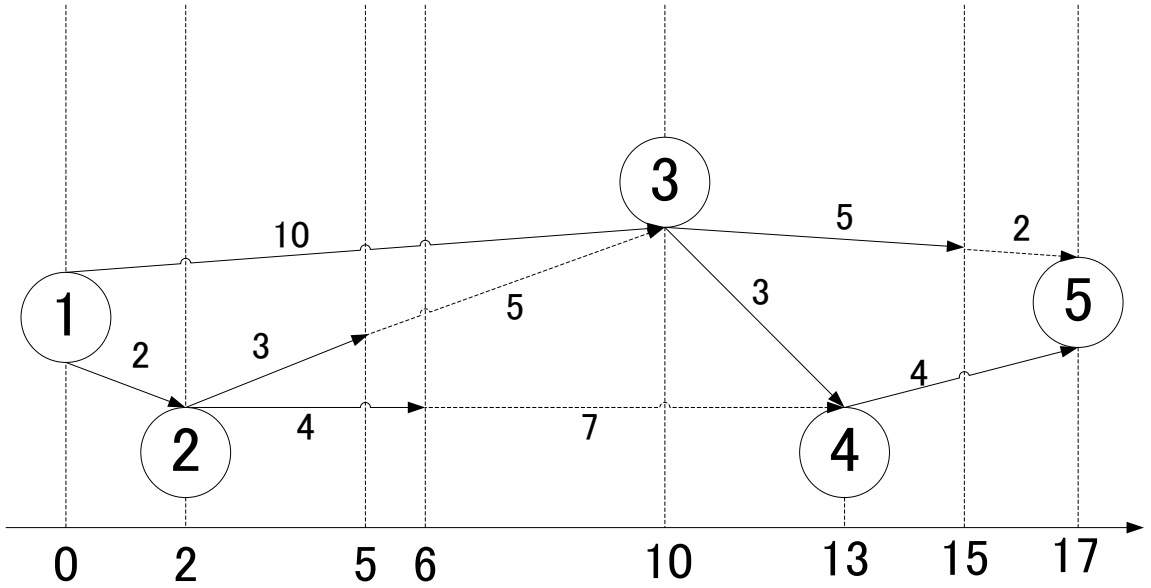
- 表のプロジェクトのアローダイヤグラムを作成し、最早開始時刻とプロジェクトの所要日数を求めよ。



アローダイヤグラム

節点番号	最早開始時刻
2	4
3	6
4	9
5	11

課題2の回答



節点番号	v_i^e	v_i^l
1	0	0
2	2	7
3	10	10
4	13	13
5	17	17

作業名	全余裕	自由余裕
1,2	5	0
2,3	5	5
2,4	7	7
3,5	2	2

組み合わせ最適化問題(1)

- 概要
  - 解ベクトルの各要素が離散的な値しかとらず、実行可能解の数が有限個に制限されている問題
  - 例：最短路問題、枝数をmとすれば解ベクトルは高々＿＿＿＿＿＿個しかない
  - 変数が整数のみを取ると言う条件で最適化を行う組み合わせ最適化法の一つ＝＿＿＿＿＿＿＿＿

組み合わせ最適化問題(2)

- 問題:
  - 変数の数が増えたときに解の数が爆発的に増えること
  - 例:0,1のみをとりうるn個の変数
    - n=10→解の数＝_____
    - n=30→解の数≒_____
    - n=50→解の数≒_____
- 効率的なアルゴリズムの構築が必要
  - 手法:問題の性質や構造を利用

組み合わせ最適化問題(3)

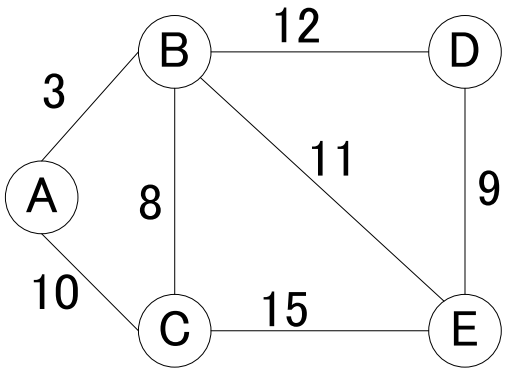
- 欲張り法
  - アルゴリズムの途中過程で、近視眼的なローカル最適化を行うことで、問題全体のグローバル最適化を行おうとする方法の総称
  - 欲張り法の適用範囲は限られている
- 分枝限定法
  - 解空間の全ての候補を総当りで検索する際に、不必要な枝を先に刈り取ることで、計算量を削減する方法
- 動的計画法 (dynamic programming)
  - 問題全体の解を求めるために、小さい問題の最適解を求め、次第に拡張していく方法
- 近似解法 (heuristic methods)
  - 最適解の探索が困難な問題に利用する方法

欲張り法（1）

- アルゴリズムの途中過程で、近視眼的なローカル最適化を行うことで、問題全体のグローバル最適化を行おうとする方法
  - この方法の適用範囲は限られている
- 必ず最適解が求まる例：
  - 最小木問題 (minimum spanning tree problem)
    - あるグラフにおける最小の長さをもち、全ての節点を含む木を求めよ

欲張り法(2)

- 問題 4.1
  - 下図のネットワークの最小木を求めよ。



最小木問題における欲張り法＝＿＿＿＿＿＿＿＿

0. 枝を長さの短い順でソートする。枝$e_i \in E$の長さをa_iとすると

$$a_1 \leq a_2 \leq \cdots \leq a_m$$

となる。$T=\{e_1\}$, $k=2$とする。

1. $T \cup \{e_k\}$が閉路を含まないときに$T \leftarrow T \cup \{e_k\}$とする。
2. Tがスパニングツリーになっていれば終了、そうでなければ$k \leftarrow k+1$として、1に戻る。

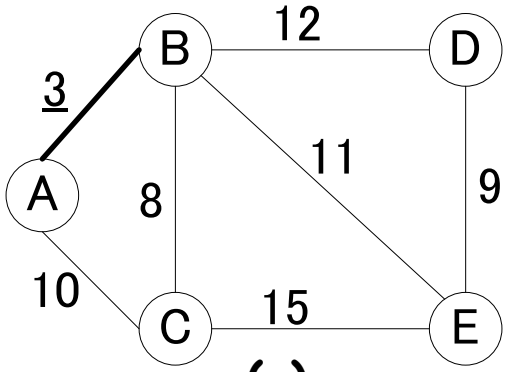
欲張り法（3）

- スパニングツリーの枝の本数は必ず＿＿＿本となるので、2. の判定は集合Tに含まれる枝数を数えるだけでよい。

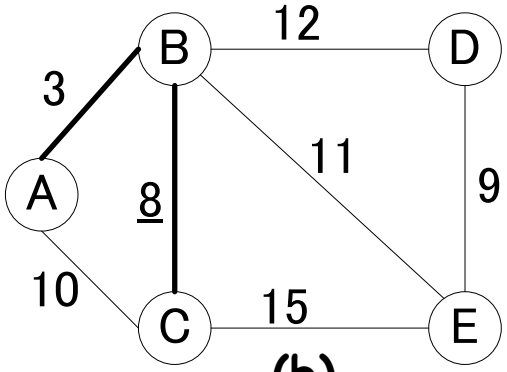
- 次ページで実際に問題4.1を解く

欲張り法（4）

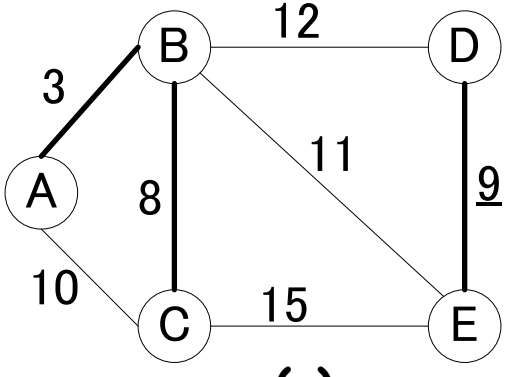
- Step 0
  - {e1, …, e7} = {(A,B), (B,C), (D,E), (A,C), (B,E), (B, D), (C,E)}



(a)

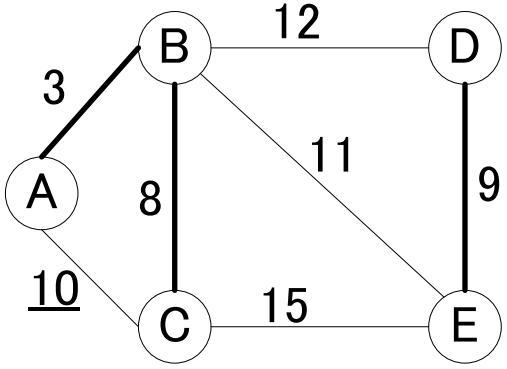


(b)

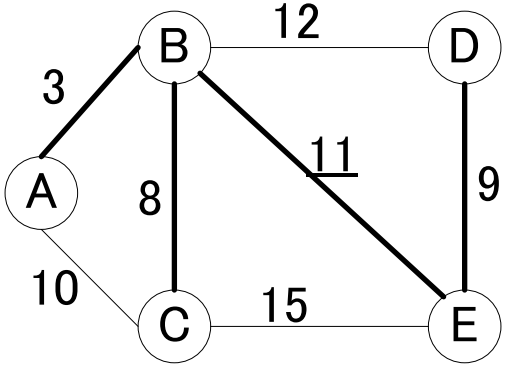


(c)

枝(A,C)を選択していない！ →



(d)



(e)

ここで枝数＝4

欲張り法（５）

- クラスカル法によって最小木が得られることを証明

全域森

ある与えられた（必ずしも連結していない）グラフにおいて、どの枝を加えても閉路が出来てしまう枝の集合を全域森という。一つのグラフにおいて全域森に属する枝の数は必ず n-n'本となる。ただし、n'は全域森中の連結グラフ数。

- クラスカル法で得られる枝の集合Tは、閉路を含まず枝の本数はn-1なので、Tは＿＿＿＿である。T内の枝をTに組み込んだ順に並べたものを $\{e_{l_1} \le e_{l_2} \le \cdots \le e_{l_{n-1}}\}$ とすると、それらは＿＿＿＿順になっている。つまり

$$a_{l_1} \le a_{l_2} \le \cdots \le a_{l_{n-1}}$$

欲張り法（６）

- 次に与えられたグラフG=(V,E)に対する任意のスパニングツリーT*を考え、Tとは別に存在すると仮定
  - T*内の枝を長さの短い順に並べる→

$$a_{q_1} \le a_{q_2} \le \cdots \le a_{q_{n-1}} \qquad (4.1)$$

  - 仮定により下式を満たすr (1 < r < n-1)が存在

$$a_{l_1} \le a_{q_1}$$
$$\vdots$$
$$a_{l_{r-1}} \le a_{q_{r-1}}$$
$$a_{l_r} > a_{q_r} \qquad (4.2)$$

（ただし、枝の選択方法よりr≠1）

欲張り法(7)

- ここで下記の枝集合T'を考える

$$T' = \{e_{l_1}, e_{l_2}, \cdots, e_{l_{r-1}}, e_{q_1}, \cdots, e_{q_r}\}$$

  - T'には枝が重複して含まれる可能性があり、グラフG'は連結グラフとは限らない
  - クラスカル法におけるTの構成法により、枝集合 $\{e_{l_1}, e_{l_2}, \cdots, e_{l_{r-1}}\}$ は部分グラフG'に対する全域森になっている。
  - なぜなら、そうでないとすると、(4.1), (4.2)式よりクラスカル法の計算仮定で e_{l_r} のかわりにそれよりも短い$e_{q_1}, \cdots, e_{q_r}$ のいずれかを付け加えたはずである。
  - 枝集合 $\{e_{l_1}, e_{l_2}, \cdots, e_{l_{r-1}}\}$がG'の全域森であることから、G'はr本以上の枝を持つ森を含まない
  - 一方 $\{e_{q_1}, \cdots, e_{q_r}\}$ はGに対する全域木T*の一部分を構成する森であり、そのあだ数はrであるから矛盾する
  - すなわち、(4.2)式の仮定は誤りであり、Tは最小木となる

ナップザック問題（1）

- n個のアイテムはそれぞれ重さがaiであり、利用価値がciである。ナップザックには重さbまでしかいれられないとき、利用価値の和を最大にするにはどうすればよいか。

目的関数： $\sum_{i=1}^{n} c_i x_i \rightarrow$ 最大化

制約条件： $\sum_{i=1}^{n} a_i x_i \leq b$

$x_i \in \{0,1\}$ ← ナップザックにものを入れるとき1

ナップザック問題（2）

- この問題を欲張り法で解くことを考える
  - まず、アイテムを「単位重さあたりの利用価値」の順に並べる

$$\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \cdots \geq \frac{c_n}{a_n}$$

  - 欲張り法で解くには__________________________ものから順に重さの制限が許す限りいれていけばよい。

ナップザック問題（3）

- 下記の問題を考える。

目的関数： $7x_1 + 8x_2 + x_3 + 2x_4 \rightarrow$ 最大化

制約条件： $4x_1 + 5x_2 + x_3 + 3x_4 \leq 6$

$x_i \in \{0,1\}$

- クラスカル法の解＝(1,0,1,0)，r=8
- 最適解＝(0,1,1,0), r=9

欲張り法の弱みと強み

- 最小木問題では、最小化するために短い枝から順にとれいれる方法（欲張り法）によって最適解の探索が可能
- ナップザック問題をはじめとする、多くの問題では最適解の探索が不能
- しかし、考え方が簡単で計算量が少ないため、近似解を得るために使われることがある