Perfect Graph
理想グラフ

点が中心のグラフ理論.

(これまでは枝が中心).

無向グラフ $G=(V,E)$

- $\omega(G)$: 最大 clique サイズ

$S(\subseteq V)$ が clique

$\overset{\text{def}}{\Leftrightarrow}$ S の全ての点が枝で結ばれている.

- $\alpha(G)$ 最大安定集合サイズ

$S(\subseteq V)$: stable set

$\overset{\text{def}}{\Leftrightarrow}$ S のどの二点も枝で結ばれていない.

(独立点集合ともいう).

- $\chi(G)$: chromatic ~~num~~ number

(min ~~col~~ coloring)

彩色数

- $\kappa(G)$ 最小クリークカバーのサイズ

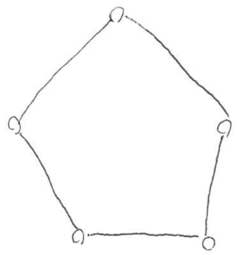
V の分割 $\{S_1, \ldots S_k\}$ で, 各 S_i が clique.

$\Updownarrow$

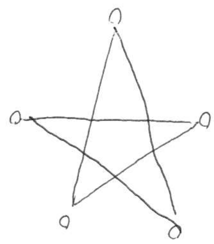
$\bigcup_{i=1}^{k} S_i = V \quad S_i \cap S_j = \emptyset \ (i \neq j)$.

Four Color Th.

平面地図は 4 - colorable



$G = C_5$



$\bar{G} = \overline{C_5}$

$$\omega(G) = 2 < 3 = \chi(\bar{G})$$

$$\alpha(G) = 2 < 3 = \kappa(G)$$

Lemma 1.

$$\omega(G) \leq \chi(G)$$

$$\alpha(G) \leq \kappa(G)$$

Lemma 2

$$\omega(G) = \alpha(\bar{G})$$

$$\chi(G) = \kappa(\bar{G})$$

$G(V, E)$, $S(\subseteq V)$

$\downarrow$ 点誘導部分グラフ

$(S, E \cap (S \times S))$

<u>Def</u>

G: χ-Perfect

$\Leftrightarrow \forall S(\subseteq V) \quad \omega(G_S) = \chi(G_S)$.

G: α-perfect

$\Leftrightarrow \forall S(\subseteq V) \quad \alpha(G_S) = K(G_S)$

<u>Lemma 3</u>

G: χ-perfect $\Leftrightarrow$ $\overline{G}$: α-perfect

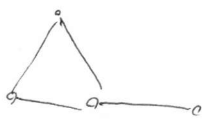
$\because$) from lemma 2.

<u>Weak Perfect Th.</u>

G: χ-perfect $\Leftrightarrow$ G: α-perfect.

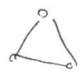
単に G:perfect

$\Updownarrow$

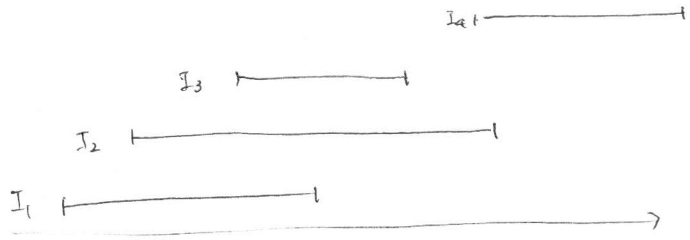


$$\max \quad x_1 + x_2 + x_3 + x_4 = \alpha(G)$$

$$\text{s.t.} \quad \begin{cases} x_1 + x_2 + x_3 \leq 1 \longleftarrow \\ x_3 + x_4 \leq 1 \longleftarrow \\ x_i \geq 0 \end{cases}$$



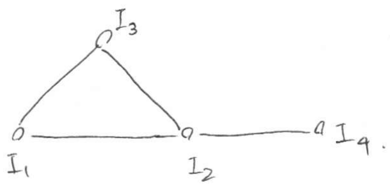
Interval Graph.



$I_i = [x_i^-, x_i^+]$. $(i=1,\cdots,n)$. are given.

各 I_i をグラフの点とし

$$I_i \cap I_j \neq \emptyset \iff \exists \text{枝} (u_i, v_j)$$
$$(i \neq j).$$



sweapline algorithm

x_i^-, x_i^+ $(i=1,\cdots,n)$ が相異なるとする.

・$\omega(G)$.

1. counter = 0　x_i^-, x_i^+ を sort.
2. sweapline を左から右に動かし、x_i^-, x_i^+ を、hit した時点を順に考える.

   x_i^- に hit → counter = counter + 1.

   x_i^+ に hit → counter = counter − 1.

3. 各時点の counter の最大値 = $\omega(G)$.

・$\chi(G_I)$

1. used_colors $= \phi$, x_i^-, x_i^+ を sort.

2. sweapline を 左から右に動かし, x_i^-, x_i^+ を hit した順に考える.

   ・x_i^- を hit.

   if used_color $\neq \phi$ then
   その中の 1色を使って I_i をぬり, used_color からその色を削除.
   else
   新しい色を導入して I_i をぬる.

   ・x_i^+ を hit.
   I_i の色を used_color に入れる.

・$\alpha(G_I) = K(G_I)$

1. x_i^-, x_i^+ を sort

2. sweapline を動かす.

   x_i^+ を hit
   → I_i を stable set に採用.
   今 sweapline と交わっている他の区間を消去.

   (今 sweapline と交わる区間集合を一つの clique として出力, reset).

最大重みのときは Dynamic Programming を使っている?

試験 9/20 (火) 10:15 ~ 11:45

グラフの（線形）代数的構造

有向グラフ $G=(V,E)$

$\rightarrow$ 接続行列 H

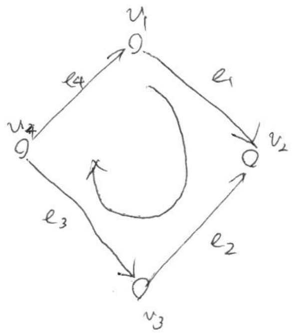
「Hの列ベクトルの一次独立性，一次従属性」

<u>Th</u> $E'(\subseteq E)$：一次独立（Hで）

$\Leftrightarrow$ E'で矢印を無視したとき，E'が閉路を含まない．

（：）

$\Rightarrow$）



このグラフを無向化．

$$\begin{array}{ccccccc} a_1 & - & a_2 & - & a_3 & + & a_4 = \mathbf{0} \\ \shortparallel & & \shortparallel & & \shortparallel & & \shortparallel \\ \times(+1) & & \times(-1) & & \times(-1) & & \times(+1) \end{array}$$

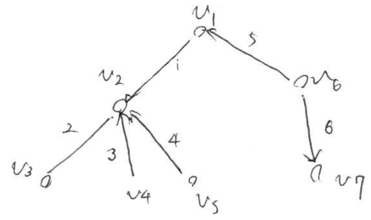
$$\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

正　逆

⇐) DFS (Depth First Search)

閉路なし → 各連結成分は木

"無向木"

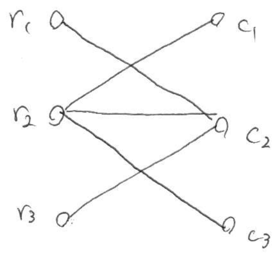


$$
\begin{array}{c|c}
 & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\
\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix} &
\begin{bmatrix}
1 & & & & -1 & \\
-1 & 1 & -1 & -1 & & \\
 & -1 & & & & \\
 & & 1 & & & \\
 & & & 1 & & \\
 & & & & 1 & 1 \\
 & & & & & -1
\end{bmatrix}
\end{array}
$$

上三角行列

対角要素 ±1.

二部グラフ → 複数個の不定元をもつ行列.



疎な構造

$$\begin{array}{c|ccc} & c_1 & c_2 & c_3 \\ \hline r_1 & & & \\ r_2 & & & \\ r_3 & & & \end{array} \quad \begin{bmatrix} 0 & x_{12} & 0 \\ x_{21} & x_{22} & x_{23} \\ 0 & x_{32} & 0 \end{bmatrix} =: A$$

det $A = 0$. 正則でない.

行ベクトル $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ の一次独立性.

↓ベクトル集合

$\{\emptyset, \{\mathbf{r}_1\}, \{\mathbf{r}_2\}, \{\mathbf{r}_3\}, \{\mathbf{r}_1, \mathbf{r}_2\}, \{\mathbf{r}_2, \mathbf{r}_3\}\}$

$\{\mathbf{r}_1, \mathbf{r}_3\}$, $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$: 一次従属.

Matroid = "Matrix" + "oid"
もどき

E: 有限台集合; $\mathcal{I}$ $(\subseteq 2^E)$: 独立集合族.

Def $(E, \mathcal{I})$: Matroid.

(I1) $\emptyset \in \mathcal{I}$

(I2) $X \subseteq Y$, $Y \in \mathcal{I} \Rightarrow X \in \mathcal{I}$

(I3) $X, Y \in \mathcal{I}$; $|X| + 1 = |Y|$

$\Rightarrow \exists y \in Y \setminus X$, $X \cup \{y\} \in \mathcal{I}$

Macroid の例

(1) 体 $\mathbb{F}$ 上の行列 A.

列ベクトル集合の一次独立性.

$\Rightarrow$ Linear Matroid over $\mathbb{F}, \mathbb{R}, \mathbb{C}$.

(2) グラフ, 閉路の有無

$\Rightarrow$ Graphic Matroid on E (edge set).

(3) 二部グラフ, 不完元

$\Rightarrow$ Transversal Matroid on 行ベクトル集合.

(4) 一様マトロイド

$U_{n,k}$

n 要素集合上で k 要素以下独立.

$U_{4,2}$

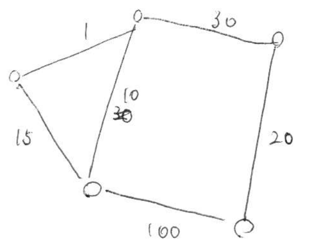
× Graphic

○ Transversal.
$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ x_{41} & x_{42} \end{bmatrix}$$

$M \rightarrow \underset{\text{dual}}{M^*}$ （直交補空間）

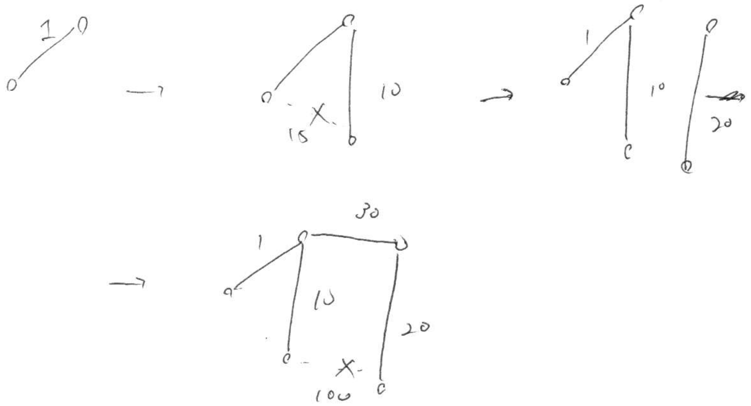
Gready Algorithm

Min - Weight Spanning Tree



① 重み小さい順 sort

② φスタート，その順に枝添加．ただし，閉路ができる場合は無視．



Kruskal

<u>Th.</u> max-weight independent set
(weight ≥ 0)

(1) Matroid ならば Gready algo. でそれが求まる.

(2) 全ての非負重みに対して Gready algo. が成功する. → Matroid.

lexicographically maximum

辞書式順序で最大

着目している独立集合の重み.

→ 大きい順に sort ⇒ ベクトル

I3 を使って. //.