

List Coloring

$v \in V(G)$, $L(v)$ a list of colors

A **list coloring** is a proper coloring f of G such that $f(v) \in L(v)$ for all $v \in V(G)$.

G is **k -choosable** or **k -list-colorable** if **every** assignment of k -element lists permits a proper coloring.

$$\chi_l(G) = \min\{k : G \text{ is } k\text{-choosable}\}$$

Claim $\chi_l(G) \geq \chi(G)$

Claim $\chi_l(G) \leq \Delta(G) + 1$

Example: $K_n, K_{2,2}$

Example: $\chi_l(K_{3,3}) \neq \chi(K_{3,3})$

Example: $\chi_l(G) - \chi(G)$ arbitrary large

Proposition $K_{m,m}$ is not k -choosable for $m = \binom{2k-1}{k}$.

1

Edge-List Coloring

List Coloring Conjecture (1985) $\chi'_l(G) = \chi'(G)$

Theorem (Galvin, 1995) $\chi'_l(B) = \chi'(B)$ for any bipartite graph B .

Proof for $B = K_{n,n}$ (Dinitz Conjecture, 1979)

2

Detour: Stable Matchings

Bonnie and Clyde is called an **unstable pair** if

- Bonnie and Clyde are currently not a couple,
- Bonnie prefers Clyde to her current partner, and
- Clyde prefers Bonnie to his current partner.

A perfect matching (of n woman and n man) is a **stable matching** if it yields no unstable pair.

Theorem. (Gale-Shapley, 1962) There exists a divorce-free society. More precisely: For any preference rankings of n man and n woman there is a stable matching.

Proof. Algorithmic.

3

The proof of divorce-free society

Proposal Algorithm (Gale-Shapley, 1962)

Input. Preference ranking by each of n man and n woman.

Iteration.

Each man **proposes** to the woman highest on his list who has **not** previously **rejected** him.

IF each woman receives exactly one proposal, THEN **stop** and **report** the resulting matching as *stable*.

ELSE

every woman receiving more than one proposal **rejects** all of them except the one highest on her list.

Every woman receiving at least one proposal says **"maybe"** to the most attractive proposal she received.

Iterate.

Theorem. The Proposal Algorithm produces a stable matching.

4

Kernel-perfect digraphs and choosability_____

A **kernel** of a digraph D is an independent set $S \subseteq V(D)$, such that for every $v \in V(D) \setminus S$ there is $w \in S$, such that $v\vec{w}$.

A digraph is **kernel-perfect** if every induced subdigraph has a kernel.

Let $f : V(G) \rightarrow N$ be a function. A graph G is called **f -choosable** if a proper coloring can be chosen from any family of lists $\{L(v)\}_{v \in V(G)}$ provided $|L(v)| \geq f(v)$ for every $v \in V(G)$.

Lemma (Bondy-Boppana-Siegel) Let D be a kernel-perfect orientation of G . Then G is f -choosable with $f(v) = 1 + d_D^+(v)$.

Theorem (Galvin, 1995) $\chi'_i(K_{n,n}) = \chi'(K_{n,n})$.

Proof. Give a kernel-perfect orientation to $L(K_{n,n})$ with $\Delta^+ = n - 1$.

5

Kernel-perfect orientation of $L(K_{n,n})$ _____

$$M = W = \{0, 1, 2, \dots, n - 1\}$$

$$E(K_{n,n}) = V(L(K_{n,n})) = \{ij : i \in M, j \in W\}$$

$$ij \rightarrow i'j \quad \text{iff} \quad i + j > i' + j \pmod{n}$$

$$ij \rightarrow ij' \quad \text{iff} \quad i + j < i + j' \pmod{n}$$

$$d^+(ij) = n - 1 \text{ for every } ij \in V(L(K_{n,n}))$$

Why do we have a kernel for every $S \subseteq V$?

Define an appropriate preference list based on S , such that for any stable matching K , $K \cap S$ is a kernel.

Man i prefers woman j to woman j' iff

$$ij \in S, ij' \in S \text{ and } ij \leftarrow ij' \text{ or}$$

$$ij \in S, ij' \notin S \text{ or}$$

$$ij \notin S, ij' \notin S \text{ and } ij \leftarrow ij'$$

6

Woman j prefers man i to man i' iff

$$ij \in S, i'j \in S \text{ and } ij \leftarrow i'j \text{ or}$$

$$ij \in S, i'j \notin S \text{ or}$$

$$ij \notin S, i'j \notin S \text{ and } ij \leftarrow i'j$$

Claim. $K \cap S$ is a kernel for $L(K_{n,n})[S]$

Proof. K is a matching $\Rightarrow K \cap S$ is independent

Suppose there is $ij \in S \setminus K$ which has no outneighbor in $K \cap S$. Let $ij', i'j \in K$.

Then either $ij' \notin S$, or $ij' \in S$ and $ij \leftarrow ij'$. In any case i prefers j to j' .

Similarly either $i'j \notin S$ or $i'j \in S$ and $ij \leftarrow i'j$. In any case j prefers i to i' .

Hence ij is an unstable pair, a contradiction.