Maximum Cut and Local Search

[Hromcovic Section 4.3.3 or Ausielle et al. Section 2.3.1]

Consider a graph $G = (V, E)$ we look for a set $S \subseteq V$ which maximizes

$\text{cost}(S) := |\{\{s,t\} \in E : s \in S, t \in V \setminus S\}|$
Local Search Algorithm

cut size: \( \text{cost}(S) := | \{ \{s, t\} \in E : s \in S, t \in V \setminus S \} | \)
symmetric difference: \( A \Delta B := (A \setminus B) \cup (B \setminus A) \)

Function \text{maximalCut}(G)

\begin{align*}
S &:= \emptyset \\
\text{while } &\exists v \in V : \text{cost}(S) < \text{cost}(S \Delta \{ s \}) \\
S &:= S \Delta \{ s \} \\
\text{return } S
\end{align*}

Running time \( O(|E|^2) \) (bug in Hromcovic?)

Exercise: Running time \( O(|V| \cdot |E|) \) (additional data structure)

Exercise: even better time bounds?
Approximation Guarantee

**Theorem 1.** \( c := \text{cost}(\text{maximalCut}(G)) \geq \frac{1}{2} \text{opt} \)

**Proof.** Let
\[
c(v) := |\{\{u, v\} : u, v \text{ are on different sides of the cut}\}|
\]
\[
c(v) \geq \frac{\text{degree}(v)}{2}.
\]
Otherwise, move \( v \) to the other side.

Thus,
\[
c = \frac{\sum_{v \in V} c(v)}{2} \geq \frac{\sum_{v \in V} \text{degree}(v)}{2} = \frac{2|E|/2}{2} = \frac{|E|}{2} \geq \frac{\text{opt}}{2}.
\]
Just flip coins...
More on Maximum Cut

[Goemanns Williamson 95]

0.87 approximation based on randomized rounding of a semidefinite programming relaxation
More on Local Search

- Simple algorithms

- Difficult runtime analysis, e.g., weighted max cut? Simplex algorithm?

- The neighborhood is an important, application dependent choice

- Allow restarts

- The real fun starts when we allow detrimental steps
  - simulated annealing
  - random search
  - threshold acceptance
  - Taboo search
  - Kernighan-Lin type heuristics
Edge Coloring

Aus Reinhard Diestel, Graphentheorie, Kapitel 4.3
http://www.math.uni-hamburg.de/home/diestel/books/graphentheorie/download.html

und J. Misra, D. Gries, A constructive proof of Vizing’s Theorem, IPL 41, p 131–133, 1992

Consider an undirected Graph $G = (V,E)$. Let $\Delta := \text{maximum degree of a vertex in } V$
$c : E \to \{1, \ldots, q\}$ is a $q$-coloring if
$\forall e \in E, e' \in E : e \neq e', e \cap e' \neq \emptyset, c(e) = c(e')$,
i.e., incident edges have different colors

$\chi'(G) := \min \{ q : \exists q\text{-edge coloring for } G \}$
Applications

- Typical iteration in **Numerics**:

  ```
  foreach \{v, w\} \in E \text{ do in arbitrary order}
  \quad (v, w, \{v, w\}) := f(v, w, \{v, w\})
  ```

  Edges in a color class can be iterated in parallel.

- Printed Circuit Board Testing: Nodes = nets; edges = potential short circuits; edges in a color class can be tested in one shot.

- File transfer: nodes = computers in a network; edges = files; files in a color class can be transmitted simultaneously

Often, we would rather **color vertices** but this is a very difficult problem.
Fast Coloring with $2\Delta - 1$ colors

\begin{verbatim}
for c := 1 to $\infty$ while $\exists$ uncolor edge do
    find a maximal matching $M$
    color all edges in $M$ with color $c$
    $E := E \setminus M$
\end{verbatim}

Exercise: Implementation in time $O(|E|)$

Exercise: Prove that $2\Delta - 1$ colors suffice

Exercise: (Near) Linear time algorithm for coloring with $3 \lceil \Delta/2 \rceil$ colors?
Theorem 2. [Vizing 64]
\[ \chi' = \Delta \text{ or } \chi' = \Delta + 1. \]
Moreover, a \( \Delta + 1 \) coloring can be found in time \( O(|V| \cdot |E|) \)

Theorem 3. [Holyer 81]
Deciding whether \( \chi' = \Delta \) or \( \chi' = \Delta + 1 \) is NP-hard.
Some Tools

Fix $q := \Delta + 1$ available colors.

Consider partial coloring $c \leftrightarrow \{1, \ldots, q\}$

Observation: every node has missing colors
\( \alpha/\beta \)-Paths and shifts

missing colors

shift
Rotating Fans

Sanders/van Stee: Approximations- und Online-Algorithmen
Constructing a Maximal Fan at \( \{X, Y\} \)

\[ d := \text{a color missing at } X \]
\[ F := \langle Y \rangle \]

while \( \exists (X, v) \in E \text{ with color } d : v \notin F \) do

\[ F := F \cup \{v\} \]
\[ d := \text{a color missing at } v \]

---

\( X \) \hspace{1cm} \( X \)

\( f \) \hspace{1cm} \( f \)

\( g \) \hspace{1cm} \( g \)

\( l \) \hspace{1cm} \( l \)

\( h \) \hspace{1cm} \( h \)
Algorithm Minus Special Cases

foreach \( \{X,Y\} \in E \) do // color it
\[
\langle f..\ell \rangle := \text{a maximal fan at } \{X,Y\} \\
c := \text{a color missing at } X \\
d := \text{a color missing at } \ell \\
\text{shift the } cd\text{-path at } X
\]

rotate fan \( \langle f..\ell \rangle \)

\[
\text{color } \{X,\ell\} \text{ with color } d
\]
Special Cases

- The fan $\langle f..\ell \rangle$ can be rotated immediately.
  - no problem. Interpret a $cc$-path as empty.

- Where does the $cd$-path at $X$ go?
Algorithm

foreach \( \{X, Y\} \in E \) do  
  \( \langle f..\ell \rangle := \) a maximal fan at \( \{X, Y\} \)  
  \( c := \) a color missing at \( X \)  
  \( d := \) a color missing at \( \ell \)  
  shift the \( cd \)-path at \( X \)  
  rotate fan \( \langle f..w \rangle : d \) missing at \( w \)  
  color \( \{X, w\} \) with color \( d \)
Multigraph Edge Coloring

Back to file transfer: What if files have different lengths or multiple files are to be transferred?

\[ \sim \text{ allow parallel edges} \]

\[ \Delta + 1 \] colors are no longer sufficient