Approximation and hardness results for the maximum edge $q$-coloring problem

Alex Popa
University of Bristol

Joint work with Anna Adamaszek and Ola Svensson

University of Liverpool
Maximum edge $q$-coloring
Problem definition
Motivation
Previous work

Our results

Approximation for graphs with a perfect matching

Hardness results

49/25-approximation algorithm

Conclusions and open problems
Maximum edge $q$-coloring

- Input: an integer $q$ and a simple, undirected graph $G = (V, E)$
Maximum edge $q$-coloring

- **Input**: an integer $q$ and a simple, undirected graph $G = (V, E)$

- **Edge $q$-coloring**: $\forall v \in V$ all the edges incident to $v$ are colored with at most $q$ different colors.
Maximum edge $q$-coloring

- Input: an integer $q$ and a simple, undirected graph $G = (V, E)$

- Edge $q$-coloring: $\forall v \in V$ all the edges incident to $v$ are colored with at most $q$ different colors.

- Goal: edge $q$-coloring that uses a maximum number of colors
Motivation - wireless mesh networks

- 2005, Raniwala and Chiueh
  - more NICs per computer
  - Two interface cards
  - network throughput \times 6

- How many channels can be used simultaneously by the network?
Motivation - wireless mesh networks

- 2005, Raniwala and Chiueh ⇒ more NICs per computer
Motivation - wireless mesh networks

- 2005, Raniwala and Chiueh ⇒ more NICs per computer
- Two interface cards ⇒ network throughput \times 6
Motivation - wireless mesh networks

- 2005, Raniwala and Chiueh ⇒ more NICs per computer
- Two interface cards ⇒ network throughput × 6
- How many channels can be used simultaneously by the network?
The **anti-Ramsey number** $ar(G, H) =$ the maximum number of colors used in an assignment in which every copy of $H$ in $G$ has at least two edges with the same color.
Previous work - extremal graph theory

- The *anti-Ramsey number* \( ar(G, H) \) is the maximum number of colors used in an assignment in which every copy of \( H \) in \( G \) has at least two edges with the same color.

- Number of colors in a maximum edge \( q \)-coloring = \( ar(G, K_{1,q+1}) \).
The anti-Ramsey number $ar(G, H) = \text{the maximum number of colors used in an assignment in which every copy of } H \text{ in } G \text{ has at least two edges with the same color}$

Number of colors in a maximum edge $q$-coloring $= ar(G, K_{1,q+1})$

Anti-Ramsey numbers have been studied since 1975.
Previous work - extremal graph theory

- The anti-Ramsey number $ar(G, H) = \text{the maximum number of colors used in an assignment in which every copy of } H \text{ in } G \text{ has at least two edges with the same color}$

- Number of colors in a maximum edge $q$-coloring = $ar(G, K_{1,q+1})$

- Anti-Ramsey numbers have been studied since 1975.

- The values of $ar(G, K_{1,q+1})$ are known for the following classes of graphs $G$: clique, complete bipartite graph, hypercube and product of cycles.
Maximum edge $q$-coloring is introduced by Feng, Zhang, Qu and Wang.
Maximum edge $q$-coloring is introduced by Feng, Zhang, Qu and Wang.

- 2-approximation algorithm for $q = 2$ and a $(1 + \frac{4q-2}{3q^2-5q+2})$-approximation for $q > 2$. 
Maximum edge $q$-coloring is introduced by Feng, Zhang, Qu and Wang.

- 2-approximation algorithm for $q = 2$ and a $(1 + \frac{4q-2}{3q^2-5q+2})$-approximation for $q > 2$.

- Exact solution for trees and complete graphs in the case $q = 2$
Maximum edge $q$-coloring is introduced by Feng, Zhang, Qu and Wang.

- 2-approximation algorithm for $q = 2$ and a $(1 + \frac{4q-2}{3q^2-5q+2})$-approximation for $q > 2$.

- Exact solution for trees and complete graphs in the case $q = 2$.

- The complexity for general graphs is an open problem.
Our results

Hardness (bipartite graphs):

- $(UGC)$: $1 + \frac{1}{q} - \epsilon$ for any $q \geq 2$.
- Without $(UGC)$: $1 + q - \frac{2}{(q-1)^2} - \epsilon$ for any $q \geq 3$ ($\approx 1.19$ for $q = 2$).

Approximation:
- $\frac{49}{25}$-approximation algorithm for general graphs.
- $\frac{5}{3}$-approximation algorithm for $q = 2$ for graphs with a perfect matching.
Our results

Hardness (bipartite graphs):

- (UGC): $1 + 1/q - \epsilon$ for any $q \geq 2$.
Our results

Hardness (bipartite graphs):

- (UGC): $1 + 1/q - \epsilon$ for any $q \geq 2$.
- (without UGC): $1 + \frac{q-2}{(q-1)^2} - \epsilon$ for any $q \geq 3$ ($\approx 1.19$ for $q = 2$).
Our results

Hardness (bipartite graphs):

- (UGC): $1 + 1/q - \epsilon$ for any $q \geq 2$.
- (without UGC): $1 + \frac{q-2}{(q-1)^2} - \epsilon$ for any $q \geq 3$ ($\approx 1.19$ for $q = 2$).

Approximation:

- 49/25-approximation algorithm for general graphs.
Our results

Hardness (bipartite graphs):

- (UGC): \(1 + 1/q - \epsilon\) for any \(q \geq 2\).
- (without UGC): \(1 + \frac{q-2}{(q-1)^2} - \epsilon\) for any \(q \geq 3\) (\(\approx 1.19\) for \(q = 2\)).

Approximation:

- 49/25-approximation algorithm for general graphs.
- 5/3-approximation algorithm for \(q = 2\) for graphs with a perfect matching.
The 2-approximation algorithm [Feng et al.]

1. Find a maximum matching $M$ in $G$.
2. Assign a unique color to each edge from $M$.
3. Find the connected components in the graph $(V, E \setminus M)$.
4. Color the edges inside each connected component using a new color.

Remark: the number of colors in the solution depends on the matching found at point 1.

Input: graph $G = (V, E)$
The 2-approximation algorithm [Feng et al.]

Input: graph $G = (V, E)$

1. Find a maximum matching $M$ in $G$. 
The 2-approximation algorithm [Feng et al.]

**Input:** graph $G = (V, E)$

1. Find a maximum matching $M$ in $G$.
2. Assign a unique color to each edge from $M$. 

Remark: the number of colors in the solution depends on the matching found at point 1.
The 2-approximation algorithm [Feng et al.]

**Input:** graph $G = (V, E)$

1. Find a maximum matching $M$ in $G$.
2. Assign a unique color to each edge from $M$.
3. Find the connected components in the graph $(V, E \setminus M)$.

Remark: the number of colors in the solution depends on the matching found at point 1.
The 2-approximation algorithm [Feng et al.]

**Input:** graph $G = (V, E)$

1. Find a maximum matching $M$ in $G$.
2. Assign a unique color to each edge from $M$.
3. Find the connected components in the graph $(V, E \setminus M)$.
4. Color the edges inside each connected component using a new color.
The 2-approximation algorithm [Feng et al.]

Input: graph $G = (V, E)$

1. Find a maximum matching $M$ in $G$.
2. Assign a unique color to each edge from $M$.
3. Find the connected components in the graph $(V, E \setminus M)$.
4. Color the edges inside each connected component using a new color.

Remark: the number of colors in the solution depends on the matching found at point 1.
Approximation for graphs with a perfect matching

**Definition**

A *minimal graph* is a simple graph $G = (V, M \cup T)$, $M \cap T = \emptyset$, consisting of a perfect matching $M$ and a tree $T$. 
Definition

A *minimal graph* is a simple graph $G = (V, M \cup T)$, $M \cap T = \emptyset$, consisting of a perfect matching $M$ and a tree $T$.

Analysis of the algorithm (sketch):

- The optimal coloring on minimal graphs uses less than $\frac{5}{6}n + 1$ colors.
Approximation for graphs with a perfect matching

**Definition**

A *minimal graph* is a simple graph $G = (V, M \cup T)$, $M \cap T = \emptyset$, consisting of a perfect matching $M$ and a tree $T$.

**Analysis of the algorithm (sketch):**

- The optimal coloring on minimal graphs uses less than $\frac{5}{6}n + 1$ colors.
- As the algorithm outputs a coloring with $\frac{n}{2} + 1$ colors, it gives a $\frac{5}{3}$-approximation for this class of graphs.
Approximation for graphs with a perfect matching

Definition

A minimal graph is a simple graph \( G = (V, M \cup T) \), \( M \cap T = \emptyset \), consisting of a perfect matching \( M \) and a tree \( T \).

Analysis of the algorithm (sketch):

- The optimal coloring on minimal graphs uses less than \( \frac{5}{6} n + 1 \) colors.
- As the algorithm outputs a coloring with \( \frac{n}{2} + 1 \) colors, it gives a \( \frac{5}{3} \)-approximation for this class of graphs.
- The minimal graphs are the worst case.
Approximation for a perfect matching and a forest

- \( G(V, E \setminus M) \) has \( c > 1 \) connected components. Let \( OPT \) be the number of colors in the optimal solution.

\[
\text{Alg. finds a solution with exactly } \frac{n}{2} + c \text{ colors.}
\]

\[
\text{By adding } c - 1 \text{ edges we transform the graph } G \text{ into a minimal graph } G'.
\]

\[
\text{The coloring } OPT \text{ of } G \text{ gives a coloring of } G' \text{ with at least } OPT - (c - 1) \text{ colors.}
\]

\[
\text{From the assumption: } OPT - (c - 1) < \frac{5}{6}n + 1 \Rightarrow OPT < \frac{5}{6}n + c \text{ and alg. gives a } \frac{5}{3} \text{ approximate solution.}
\]
Approximation for a perfect matching and a forest

- $G(V, E \setminus M)$ has $c > 1$ connected components. Let $OPT$ be the number of colors in the optimal solution.
- Alg. finds a solution with exactly $n/2 + c$ colors.
Approximation for a perfect matching and a forest

- $G(V, E \setminus M)$ has $c > 1$ connected components. Let $OPT$ be the number of colors in the optimal solution.
- Alg. finds a solution with exactly $n/2 + c$ colors.
- By adding $c - 1$ edges we transform the graph $G$ into a minimal graph $G'$. 
Approximation for a perfect matching and a forest

- $G(V, E \setminus M)$ has $c > 1$ connected components. Let $OPT$ be the number of colors in the optimal solution.
- Alg. finds a solution with exactly $n/2 + c$ colors.
- By adding $c - 1$ edges we transform the graph $G$ into a minimal graph $G'$.
- The coloring $OPT$ of $G$ gives a coloring of $G'$ with at least $OPT - (c - 1)$ colors.
Approximation for a perfect matching and a forest

- $G(V, E \setminus M)$ has $c > 1$ connected components. Let $OPT$ be the number of colors in the optimal solution.
- Alg. finds a solution with exactly $n/2 + c$ colors.
- By adding $c - 1$ edges we transform the graph $G$ into a minimal graph $G'$.

- The coloring $OPT$ of $G$ gives a coloring of $G'$ with at least $OPT - (c - 1)$ colors.
- From the assumption:
  $$OPT - (c - 1) < \frac{5}{6}n + 1 \Rightarrow OPT < \frac{5}{6}n + c$$ and alg. gives a $5/3$ approximate solution.
Approximation for general graphs

Proof by contradiction:

- Let $G$ be the smallest graph on which the alg. is not a $5/3$ approximation.

![Graph diagram]

- Fix a vertex $v$ from the cycle.
- The edges incident to $v$ can have only two colors $⇒$ some colors have to repeat.
- Therefore $G$ is not minimal (contradiction).
Proof by contradiction:

- Let $G$ be the smallest graph on which the alg. is not a $5/3$ approximation.
- $G$ is not a minimal graph or a graph consisting of a perfect matching and a forest. It has some cycle consisting of non-matching edges.
Proof by contradiction:

- Let $G$ be the smallest graph on which the alg. is not a $5/3$ approximation.
- $G$ is not a minimal graph or a graph consisting of a perfect matching and a forest. It has some cycle consisting of non-matching edges.
- Fix a vertex $v$ from the cycle.
Proof by contradiction:

- Let $G$ be the smallest graph on which the alg. is not a $5/3$ approximation.
- $G$ is not a minimal graph or a graph consisting of a perfect matching and a forest. It has some cycle consisting of non-matching edges.
- Fix a vertex $v$ from the cycle.
- The edges incident to $v$ can have only two colors $\Rightarrow$ some colors have to repeat.

Alex Popa
Maximum edge $q$-coloring problem
Approximation for general graphs

Proof by contradiction:

- Let $G$ be the smallest graph on which the alg. is not a $5/3$ approximation.
- $G$ is not a minimal graph or a graph consisting of a perfect matching and a forest. It has some cycle consisting of non-matching edges.
- Fix a vertex $v$ from the cycle.
- The edges incident to $v$ can have only two colors $\Rightarrow$ some colors have to repeat.
Approximation for general graphs

Proof by contradiction:

- Let $G$ be the smallest graph on which the alg. is not a $5/3$ approximation.
- $G$ is not a minimal graph or a graph consisting of a perfect matching and a forest. It has some cycle consisting of non-matching edges.
- Fix a vertex $v$ from the cycle.
  - The edges incident to $v$ can have only two colors $\Rightarrow$ some colors have to repeat.
  - Therefore $G$ is not minimal (contradiction).
Lemma

Each edge 2-coloring of a minimal graph uses less than $\frac{5}{6}n + 1$ colors
Upper bound in a minimal graph

**Lemma**

Each edge 2-coloring of a minimal graph uses less than $\frac{5}{6}n + 1$ colors.

**Part I:** in the worst case edges colored with a single color form either a star or a 2-star.
Upper bound in a minimal graph

Lemma

*Each edge 2-coloring of a minimal graph uses less than $\frac{5}{6} n + 1$ colors*

**Part I:** in the worst case edges colored with a single color form either a star or a 2-star.

**Part II:** such a coloring cannot use more than $5n/6$ colors.
A tight example
A tight example
Hardness results

- Reduction from \( q \)-uniform hypergraph vertex cover.
Hardness results

- Reduction from $q$-uniform hypergraph vertex cover.
- $q$-uniform hypergraph $H(V, E) \rightarrow$ bipartite graph $G(V', E')$ such that:

\[ q(|V| + 1) - \text{OPT}_{VC}(H) = \text{OPT}_{MEC}(G) \]

Hardness:

- $(\text{UGC})$: $1 + 1/q - \epsilon$ for any $q \geq 2$.
- $(\text{without UGC})$: $1 + q - 2(q - 1)^2 - \epsilon$ for any $q \geq 3$ ($\approx 1.19$ for $q = 2$).
Hardness results

- Reduction from $q$-uniform hypergraph vertex cover.
- $q$-uniform hypergraph $H(V, E) \rightarrow$ bipartite graph $G(V', E')$ such that:

\[
q(|V| + 1) - \text{OPT}_{VC}(H) = \text{OPT}_{MEC}(G)
\]
Hardness results

- Reduction from \( q \)-uniform hypergraph vertex cover.
- \( q \)-uniform hypergraph \( H(V, E) \) → bipartite graph \( G(V', E') \) such that:

\[
q(|V| + 1) - \text{OPT}_{VC}(H) = \text{OPT}_{MEC}(G)
\]

Hardness:
- (UGC): \( 1 + 1/q - \epsilon \) for any \( q \geq 2 \).
Hardness results

- Reduction from $q$-uniform hypergraph vertex cover.
- $q$-uniform hypergraph $H(V, E) \rightarrow$ bipartite graph $G(V', E')$ such that:

$$q(|V| + 1) - OPT_{VC}(H) = OPT_{MEC}(G)$$

Hardness:

- (UGC): $1 + 1/q - \epsilon$ for any $q \geq 2$.
- (without UGC): $1 + \frac{q-2}{(q-1)^2} - \epsilon$ for any $q \geq 3$ ($\approx 1.19$ for $q = 2$).
Figure: Graph $G$ arising from a cycle with vertices $\{1, 2, 3, 4\}$ when $q = 2$. 
49/25-approximation algorithm

**Input:** graph $G = (V, E)$
49/25-approximation algorithm

**Input:** graph $G = (V, E)$

1. Run algorithm for graphs with a large optimum.
49/25-approximation algorithm

Input: graph $G = (V, E)$

1. Run algorithm for graphs with a large optimum.
2. Run algorithm for general graphs.
49/25-approximation algorithm

**Input:** graph $G = (V, E)$

1. Run algorithm for graphs with a large optimum.
2. Run algorithm for general graphs.
3. Return the solution using more colors.
Algorithm for graphs with large optimum

Input:

- graph \( G = (V, E) \)

1. \( V_2 \subseteq V \) = vertices of degree 2.
2. \( V'_2 \subseteq V_2 \) = vertices that have only neighbours of degree 2.
3. Color the edges incident to \( V'_2 \) with unique colors.
4. Color the remaining edges with an additional color \( F \).
Algorithm for graphs with large optimum

Input: graph $G = (V, E)$

1. $V_2 \subseteq V$ = vertices of degree 2.
Algorithm for graphs with large optimum

Input: graph $G = (V, E)$

1. $V_2 \subseteq V$ = vertices of degree 2.
2. $V_2' \subseteq V_2$ = vertices that have only neighbours of degree 2.
Algorithm for graphs with large optimum

Input: graph \( G = (V, E) \)

1. \( V_2 \subseteq V = \) vertices of degree 2.
2. \( V_2' \subseteq V_2 = \) vertices that have only neighbours of degree 2.
3. Color the edges incident to \( V_2' \) with unique colors.
Algorithm for graphs with large optimum

**Input:** graph $G = (V, E)$

1. $V_2 \subseteq V = \text{vertices of degree 2}$.
2. $V'_2 \subseteq V_2 = \text{vertices that have only neighbours of degree 2}$.
3. Color the edges incident to $V'_2$ with unique colors.
4. Color the remaining edges with an additional color $F$. 
Edmonds-Gallai decomposition

C

A

D

Maximum edge $q$-coloring problem
Algorithm for general graphs

Input: graph $G = (V, E)$

1. Color with distinct colors the edges from the matching incident with $A$ and $C$.
2. Color with $F$ the other edges incident to $A$ and $C$.
3. Good components and not incident to an edge not colored with $F$:
   3.1 No chord.
   3.2 Chord.
4. Bad components (or incident to an edge not colored with $F$).
Algorithm for general graphs

**Input:** graph $G = (V, E)$

1. Color with distinct colors the edges from the matching incident with $A$ and $C$.
Algorithm for general graphs

**Input:** graph $G = (V, E)$

1. Color with distinct colors the edges from the matching incident with $A$ and $C$.
2. Color with $F$ the other edges incident to $A$ and $C$. 
Algorithm for general graphs

**Input:** graph $G = (V, E)$

1. Color with distinct colors the edges from the matching incident with $A$ and $C$.
2. Color with $F$ the other edges incident to $A$ and $C$.
3. Good components and not incident to an edge not colored with $F$:
   3.1 No chord.
Algorithm for general graphs

**Input:** graph $G = (V, E)$

1. Color with distinct colors the edges from the matching incident with $A$ and $C$.
2. Color with $F$ the other edges incident to $A$ and $C$.
3. Good components and not incident to an edge not colored with $F$:
   3.1 No chord.
   3.2 Chord.
Algorithm for general graphs

**Input:** graph $G = (V, E)$

1. Color with distinct colors the edges from the matching incident with $A$ and $C$.
2. Color with $F$ the other edges incident to $A$ and $C$.
3. Good components and not incident to an edge not colored with $F$:
   3.1 No chord.
   3.2 Chord.
4. Bad components (or incident to an edge not colored with $F$).
Lemma

\[ \text{OPT}_{MEC}(G) \geq (1 - \epsilon)n \Rightarrow |V'_2| \geq (1 - 12\epsilon)n. \]
Lemma

\( OPT_{MEC}(G) \geq (1 - \epsilon)n \Rightarrow |V'_2| \geq (1 - 12\epsilon)n. \)

\( \chi_v = \) set of colors incident to vertex \( v \)
Lemma

$$OPT_{MEC}(G) \geq (1 - \epsilon)n \Rightarrow |V'_2| \geq (1 - 12\epsilon)n.$$ 

$$\chi_v = \text{set of colors incident to vertex } v$$

$$C_i = \text{colors incident to exactly } i \text{ vertices}$$
Analysis - large optimum

**Lemma**

\[ \text{OPT}_{\text{MEC}}(G) \geq (1 - \epsilon)n \Rightarrow |V'_2| \geq (1 - 12\epsilon)n. \]

\( \chi_v \) = set of colors incident to vertex \( v \)
\( C_i \) = colors incident to exactly \( i \) vertices

**Claim**

\[ |C_2| \geq (1 - 3\epsilon)n \]
Lemma

\[ \text{OPT}_{\text{MEC}}(G) \geq (1 - \epsilon)n \Rightarrow |V'_2| \geq (1 - 12\epsilon)n. \]

\( \chi_v \) = set of colors incident to vertex \( v \)

\( C_i \) = colors incident to exactly \( i \) vertices

Claim

\[ |C_2| \geq (1 - 3\epsilon)n \]

Proof.

Suppose otherwise. Then,
### Lemma

\[ \text{OPT}_{\text{MEC}}(G) \geq (1 - \epsilon)n \Rightarrow |V_2'| \geq (1 - 12\epsilon)n. \]

\( \chi_v \) = set of colors incident to vertex \( v \)

\( C_i \) = colors incident to exactly \( i \) vertices

### Claim

\[ |C_2| \geq (1 - 3\epsilon)n \]

### Proof.

Suppose otherwise. Then,

\[ 2n \geq \]
Analysis - large optimum

Lemma

$$OPT_{MEC}(G) \geq (1 - \epsilon)n \Rightarrow |V'_2| \geq (1 - 12\epsilon)n.$$

$$\chi_v = \text{set of colors incident to vertex } v$$

$$C_i = \text{colors incident to exactly } i \text{ vertices}$$

Claim

$$|C_2| \geq (1 - 3\epsilon)n$$

Proof.

Suppose otherwise. Then,

$$2n \geq \sum_{v \in V} |\chi_v| =$$
## Analysis - large optimum

### Lemma

\[ \text{OPT}_{\text{MEC}}(G) \geq (1 - \epsilon)n \Rightarrow |V'_2| \geq (1 - 12\epsilon)n. \]

\( \chi_v \) = set of colors incident to vertex \( v \)

\( C_i \) = colors incident to exactly \( i \) vertices

### Claim

\[ |C_2| \geq (1 - 3\epsilon)n \]

### Proof.

Suppose otherwise. Then,

\[ 2n \geq \sum_{v \in V} |\chi_v| = \sum_{i=2}^{n} i |C_i| \geq \]
### Lemma

\[ \text{OPT}_{\text{MEC}}(G) \geq (1 - \epsilon)n \Rightarrow |V'_2| \geq (1 - 12\epsilon)n. \]

\( \chi_v \) = set of colors incident to vertex \( v \)

\( C_i \) = colors incident to exactly \( i \) vertices

### Claim

\[ |C_2| \geq (1 - 3\epsilon)n \]

### Proof.

Suppose otherwise. Then,

\[ 2n \geq \sum_{v \in V} |\chi_v| = \sum_{i=2}^{n} i|C_i| \geq 2|C_2| + 3(\text{OPT}_{\text{MEC}}(G) - |C_2|) > \]
## Analysis - large optimum

### Lemma

\[ \text{OPT}_{\text{MEC}}(G) \geq (1 - \epsilon)n \Rightarrow |V'_2| \geq (1 - 12\epsilon)n. \]

- \( \chi_v \) = set of colors incident to vertex \( v \)
- \( C_i \) = colors incident to exactly \( i \) vertices

### Claim

\[ |C_2| \geq (1 - 3\epsilon)n \]

### Proof.

Suppose otherwise. Then,
\[
2n \geq \sum_{v \in V} |\chi_v| = \sum_{i=2}^{n} i|C_i| \geq 2|C_2| + 3(\text{OPT}_{\text{MEC}}(G) - |C_2|) > 2n
\]
The sum of unique colors incident to each vertex is
\[ 2|C_2| \geq (2 - 6\epsilon)n. \]
Analysis - large optimum

- The sum of unique colors incident to each vertex is
  \(2|C_2| \geq (2 - 6\epsilon)n\).
- The vertices adjacent \textit{only} to unique colors \(\geq (1 - 6\epsilon)n\).
The sum of unique colors incident to each vertex is $2|C_2| \geq (2 - 6\epsilon)n$.

The vertices adjacent *only* to unique colors $\geq (1 - 6\epsilon)n$.

The vertices adjacent to at most one unique color $\leq 6\epsilon n$. 
Analysis - large optimum

- The sum of unique colors incident to each vertex is $2|C_2| \geq (2 - 6\epsilon)n$.
- The vertices adjacent *only* to unique colors $\geq (1 - 6\epsilon)n$
- The vertices adjacent to at most one unique color $\leq 6\epsilon n$.
The sum of unique colors incident to each vertex is \(2|C_2| \geq (2 - 6\epsilon)n\).

- The vertices adjacent *only* to unique colors \(\geq (1 - 6\epsilon)n\).
- The vertices adjacent to at most one unique color \(\leq 6\epsilon n\).

\[|V'_2| \geq (1 - 12\epsilon)n\]
Conclusions and open problems

1 \pm \frac{1}{q} - \epsilon\text{ hardness for any } q \geq 2\text{ assuming the UGC}

1 + \frac{q}{2} - \frac{2}{q(q-1)} \text{ for any } q \geq 3 (\approx 1.19\text{ for } q = 2)

49/25-approximation algorithm for general graphs.

5/3-approximation algorithm for \( q = 2 \) for graphs with a perfect matching.

Is it possible to improve the lower and the upper bounds?
Conclusions and open problems

- $1 + \frac{1}{q} - \epsilon$ hardness for any $q \geq 2$ assuming the UGC
Conclusions and open problems

- $1 + \frac{1}{q} - \epsilon$ hardness for any $q \geq 2$ assuming the UGC or $1 + \frac{q-2}{(q-1)^2} - \epsilon$ for any $q \geq 3$ ($\approx 1.19$ for $q = 2$) without UGC.
Conclusions and open problems

- $1 + 1/q - \epsilon$ hardness for any $q \geq 2$ assuming the UGC or
  $1 + \frac{q-2}{(q-1)^2} - \epsilon$ for any $q \geq 3$ ($\approx 1.19$ for $q = 2$) without UGC.

- 49/25-approximation algorithm for general graphs.
Conclusions and open problems

- $1 + 1/q - \epsilon$ hardness for any $q \geq 2$ assuming the UGC or $1 + \frac{q-2}{(q-1)^2} - \epsilon$ for any $q \geq 3$ ($\approx 1.19$ for $q = 2$) without UGC.
- 49/25-approximation algorithm for general graphs.
- 5/3-approximation algorithm for $q = 2$ for graphs with a perfect matching.
Conclusions and open problems

- $1 + 1/q - \epsilon$ hardness for any $q \geq 2$ assuming the UGC or $1 + \frac{q-2}{(q-1)^2} - \epsilon$ for any $q \geq 3$ ($\approx 1.19$ for $q = 2$) without UGC.
- 49/25-approximation algorithm for general graphs.
- 5/3-approximation algorithm for $q = 2$ for graphs with a perfect matching.
- Is it possible to improve the lower and the upper bounds?
Thank you!
Mulţumesc!
Dziękuję!
Tack!


