Tight lower bounds for the complexity of multicoloring

Marthe Bonamy

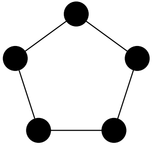
October 18th, 2016

Joint work with
Łukasz Kowalik, Michał Pilipczuk, Arkadiusz Socała, Marcin Wrochna

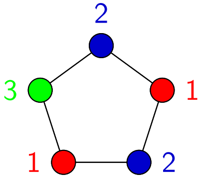
LaBRI

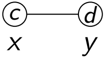
Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski

Coloring



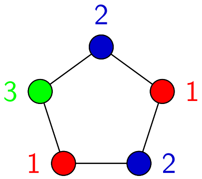
Coloring



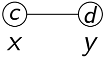


$\Rightarrow c \neq d$

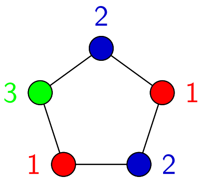
Coloring



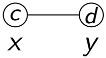
χ: Minimum number of colors to ensure that:



$\Rightarrow c \neq d$

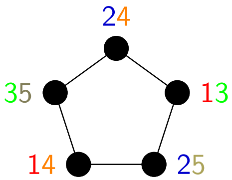


χ: Minimum number of colors to ensure that:

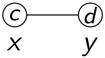


$\Rightarrow c \neq d$

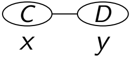
MultiColoring



χ: Minimum number of colors to ensure that:

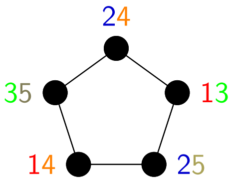


$\Rightarrow c \neq d$



$$\Rightarrow \begin{cases} |C| = b \\ |D| = b \\ C \cap D = \emptyset \end{cases}$$

MultiColoring



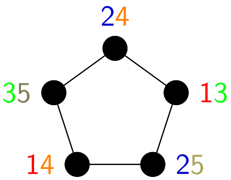
χ: Minimum number of colors to ensure that:

$$\underset{x}{\textcircled{c}} \text{———} \underset{y}{\textcircled{d}} \Rightarrow c \neq d$$

χ_b: Minimum number of colors to ensure that:

$$\underset{x}{(C)} \text{—} \underset{y}{(D)} \Rightarrow \begin{cases} |C| = b \\ |D| = b \\ C \cap D = \emptyset \end{cases}$$

MultiColoring



χ: Minimum number of colors to ensure that:

$$\underset{x}{(c)} — \underset{y}{(d)} \Rightarrow c \neq d$$

χ_b: Minimum number of colors to ensure that:

$$\underset{x}{(C)} — \underset{y}{(D)} \Rightarrow \begin{cases} |C| = b \\ |D| = b \\ C \cap D = \emptyset \end{cases}$$

$$lim_{b \to \infty} \frac{\chi_b}{b} = \chi_f$$

Complexity

"Is G k-colorable?"

Complexity

"Is G k-colorable?"

- $k \leq 1$: Easy ✓

Complexity

"Is G k-colorable?"

- $k \leq 1$: Easy ✓
- $k = 2$: Easy ✓

Complexity

"Is G k-colorable?"

- $k \leq 1$: Easy ✓
- $k = 2$: Easy ✓
- $k \geq 3$: NP-hard

Complexity

"Is G k-colorable?"

- $k \leq 1$: Easy ✓
- $k = 2$: Easy ✓
- $k \geq 3$: NP-hard

"Is G a:b-colorable?"

Complexity

"Is G k-colorable?"

- $k \leq 1$: Easy ✓
- $k = 2$: Easy ✓
- $k \geq 3$: NP-hard

"Is G $a{:}b$-colorable?"

- $a < 2b$: Easy ✓

Complexity

"Is G k-colorable?"

- $k \leq 1$: Easy ✓
- $k = 2$: Easy ✓
- $k \geq 3$: NP-hard

"Is G a:b-colorable?"

- $a < 2b$: Easy ✓
- $a = 2b$: Easy ✓

Complexity

"Is G k-colorable?"

- $k \leq 1$: Easy ✓
- $k = 2$: Easy ✓
- $k \geq 3$: NP-hard

"Is G a:b-colorable?"

- $a < 2b$: Easy ✓
- $a = 2b$: Easy ✓
- $a \geq 2b + 1$: NP-hard (Hell, Nešetril '90)

NP-hard? :(

NP-hard? :(

Exponential Time Hypothesis (Impagliazzo, Paturi '99)

There is $\epsilon > 0$ such that 3-SAT cannot be solved in $\mathcal{O}^*(2^{\epsilon \cdot n})$ time.

NP-hard? :(

Exponential Time Hypothesis (Impagliazzo, Paturi '99)

There is $\epsilon > 0$ such that 3-SAT cannot be solved in $\mathcal{O}^*(2^{\epsilon \cdot n})$ time.

Theorem (Dell, Husfeldt, Wahlén '10)

For any $k \geq 3$, there is $\alpha > 0$ such that k-Coloring cannot be solved in $\mathcal{O}^*(2^{\alpha \cdot n})$ time unless ETH fails.

NP-hard? :(

Exponential Time Hypothesis (Impagliazzo, Paturi '99)

There is $\epsilon > 0$ such that 3-SAT cannot be solved in $\mathcal{O}^*(2^{\epsilon \cdot n})$ time.

Theorem (Dell, Husfeldt, Wahlén '10)

For any $k \geq 3$, there is $\alpha > 0$ such that k-Coloring cannot be solved in $\mathcal{O}^*(2^{\alpha \cdot n})$ time unless ETH fails.

Theorem (Björklund, Husfeldt '06)

k-Coloring can be solved in $\mathcal{O}^*(2^n)$ time.

Our result

Theorem (Nederlof '08)

a:b-Coloring can be solved in $\mathcal{O}^*((b+1)^n)$ time.

Our result

Theorem (Nederlof '08)

a:b-Coloring can be solved in $\mathcal{O}^*((b+1)^n)$ time.

Theorem (B., Kowalik, Pilipczuk, Socała, Wrochna '16)

There is $\alpha > 0$ such that, for appropriate ranges of values, a:b-Coloring cannot be solved in $\mathcal{O}^*((b+1)^{\alpha \cdot n})$ time unless ETH fails.

The reduction

Fix a, b.

The reduction

Fix a, b. Main idea: compress an instance ϕ of 3-SAT on n variables and m clauses into the a:b-coloring of a graph G on $O(\frac{m+n}{\log b})$ vertices.

The reduction

Fix a, b. Main idea: compress an instance ϕ of 3-SAT on n variables and m clauses into the a:b-coloring of a graph G on $O(\frac{m+n}{\log b})$ vertices.

Sparsification Lemma (Tovey '84)

We can assume that in ϕ, every variable belongs to at most 4 clauses.

The reduction

Fix a, b. Main idea: compress an instance ϕ of 3-SAT on n variables and m clauses into the a:b-coloring of a graph G on $O(\frac{m+n}{\log b})$ vertices.

Sparsification Lemma (Tovey '84)

We can assume that in ϕ, every variable belongs to at most 4 clauses.

We can also relax a:b-coloring: every vertex is assigned

- an integer $\in \{1, \ldots, b\}$ (number of colors to receive) and
- a subset of $\{1, \ldots, a\}$ (colors it's allowed to take).

The reduction (2)

$v_1, \ldots, v_n$: variables of ϕ.
$c_1, \ldots, c_m$: clauses of ϕ.

The reduction (2)

$v_1, \ldots, v_n$: variables of ϕ.
$c_1, \ldots, c_m$: clauses of ϕ.

- Groups of variables of size $\log b$: $V_1, \ldots, V_{\frac{n}{\log b}}$
- Groups of clauses of size b: $C_1, \ldots, C_{\frac{m}{b}}$

The reduction (2)

$v_1, \ldots, v_n$: variables of ϕ.

$c_1, \ldots, c_m$: clauses of ϕ.

- Groups of variables of size $\log b$: $V_1, \ldots, V_{\frac{n}{\log b}}$
- Groups of clauses of size b: $C_1, \ldots, C_{\frac{m}{b}}$

To each group of variables, associate b colors corresponding to all possible assignments.

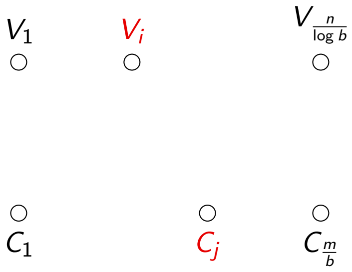
The reduction (2)

$v_1, \ldots, v_n$: variables of ϕ.

$c_1, \ldots, c_m$: clauses of ϕ.

- Groups of variables of size $\log b$: $V_1, \ldots, V_{\frac{n}{\log b}}$
- Groups of clauses of size b: $C_1, \ldots, C_{\frac{m}{b}}$

To each group of variables, associate b colors corresponding to all possible assignments.



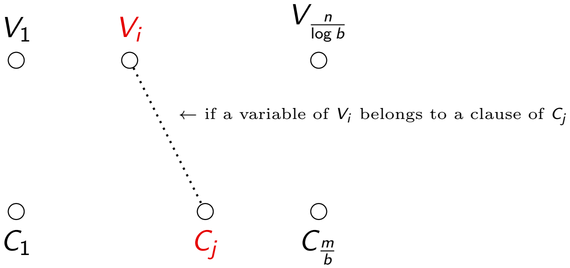
The reduction (2)

$v_1, \ldots, v_n$: variables of ϕ.

$c_1, \ldots, c_m$: clauses of ϕ.

- Groups of variables of size $\log b$: $V_1, \ldots, V_{\frac{n}{\log b}}$
- Groups of clauses of size b: $C_1, \ldots, C_{\frac{m}{b}}$

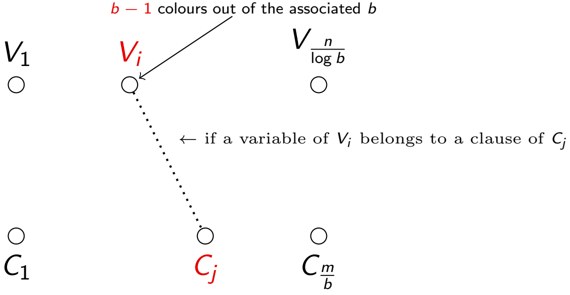
To each group of variables, associate b colors corresponding to all possible assignments.



$v_1, \ldots, v_n$: variables of ϕ.
$c_1, \ldots, c_m$: clauses of ϕ.

- Groups of variables of size $\log b$: $V_1, \ldots, V_{\frac{n}{\log b}}$
- Groups of clauses of size b: $C_1, \ldots, C_{\frac{m}{b}}$

To each group of variables, associate b colors corresponding to all possible assignments.



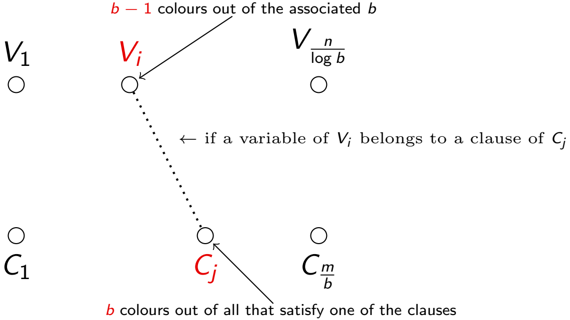
The reduction (2)

$v_1, \ldots, v_n$: variables of ϕ.

$c_1, \ldots, c_m$: clauses of ϕ.

- Groups of variables of size $\log b$: $V_1, \ldots, V_{\frac{n}{\log b}}$
- Groups of clauses of size b: $C_1, \ldots, C_{\frac{m}{b}}$

To each group of variables, associate b colors corresponding to all possible assignments.



d-detecting sets

Given a set X and a (mysterious) weight function $\omega : X \to \{-d, -d+1, \ldots, d-1, d\}$,

Minimum size of a collection $(S_1, \ldots, S_p)$ s.t.
if $\sum_{a \in S_i} \omega(a) = 0$ for every i then $\omega \equiv 0$?

d-detecting sets

Given a set X and a (mysterious) weight function $\omega : X \to \{-d, -d+1, \ldots, d-1, d\}$,

Minimum size of a collection $(S_1, \ldots, S_p)$ s.t.
if $\sum_{a \in S_i} \omega(a) = 0$ for every i then $\omega \equiv 0$?

$\rightsquigarrow$ encodes all subsets of X

d-detecting sets

Given a set X and a (mysterious) weight function $\omega : X \to \{-d, -d+1, \ldots, d-1, d\}$,

Minimum size of a collection $(S_1, \ldots, S_p)$ s.t.
if $\sum_{a \in S_i} \omega(a) = 0$ for every i then $\omega \equiv 0$?

$\rightsquigarrow$ encodes all subsets of $X \Rightarrow p \geq \frac{|X|}{\log |X|}$.

d-detecting sets

Given a set X and a (mysterious) weight function $\omega : X \to \{-d, -d+1, \ldots, d-1, d\}$,

Minimum size of a collection $(S_1, \ldots, S_p)$ s.t.
if $\sum_{a \in S_i} \omega(a) = 0$ for every i then $\omega \equiv 0$?

$\rightsquigarrow$ encodes all subsets of $X \Rightarrow p \geq \frac{|X|}{\log|X|}$.

$O(\frac{|X|}{\log|X|})$ is enough! (Lindström '65)

Conclusion

Conclusion

Thanks!

Homomorphism

Definition

A graph G is homomorphic to a graph H if there is a function $f : V(G) \to V(H)$ that preserves adjacency.

Homomorphism

Definition

A graph G is homomorphic to a graph H if there is a function $f : V(G) \to V(H)$ that preserves adjacency.

k-coloring: homomorphism to K_k.

Homomorphism

Definition

A graph G is homomorphic to a graph H if there is a function $f : V(G) \to V(H)$ that preserves adjacency.

k-coloring: homomorphism to K_k.
a:b-coloring: homomorphism to $KG_{a,b}$.
(graph on vertex set $\binom{\{1,\ldots,a\}}{b}$ with edges between disjoint sets).

Homomorphism

Definition

A graph G is homomorphic to a graph H if there is a function $f : V(G) \to V(H)$ that preserves adjacency.

k-coloring: homomorphism to K_k.
a:b-coloring: homomorphism to $KG_{a,b}$.
(graph on vertex set $\binom{\{1,\ldots,a\}}{b}$ with edges between disjoint sets).

Theorem (Hell, Nešetril '90)

For fixed H, "is G homomorphic to H?": NP-hard unless H is bipartite.

Homomorphism

Definition

A graph G is homomorphic to a graph H if there is a function $f : V(G) \rightarrow V(H)$ that preserves adjacency.

k-coloring: homomorphism to K_k.
a:b-coloring: homomorphism to $KG_{a,b}$.
(graph on vertex set $\binom{\{1,\ldots,a\}}{b}$ with edges between disjoint sets).

Theorem (Hell, Nešetril '90)

For fixed H, "is G homomorphic to H?": NP-hard unless H is bipartite.

Theorem (Cygan et al '16)

"is G homomorphic to H?" cannot be solved in $\mathcal{O}^*(|V(H)|^{\alpha \cdot |V(G)|})$ time unless ETH fails.