Parameterized and Approximation Algorithm for the c-LOAD COLORING problem

F. Barbero avec G. Gutin, M. Jones et B. Sheng

ROYAL HOLLOWAY UNIVERSITY OF LONDON

JGA - Mercredi 4 novembre 2015

The c-Load Coloring problem

c-Load Coloring

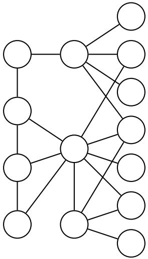
Input : graph $G = (V, E)$; integer $k \geq 0$ as parameter

Output : $\exists ?\ \varphi : V \to [c], \forall i \in [c], |E(\varphi^{-1}(i))| \geq k$

The c-Load Coloring problem

c-Load Coloring

> **Input :** graph $G = (V, E)$; integer $k \geq 0$ as parameter
> **Output :** $\exists ?\ \varphi : V \to [c], \forall i \in [c], |E(\varphi^{-1}(i))| \geq k$

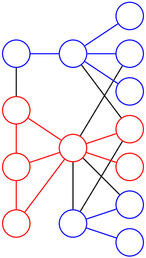


(G, k) for 2-Load Coloring ?

The c-Load Coloring problem

c-Load Coloring

Input : graph $G = (V, E)$; integer $k \geq 0$ as parameter
Output : $\exists ?\ \varphi : V \to [c], \forall i \in [c], |E(\varphi^{-1}(i))| \geq k$



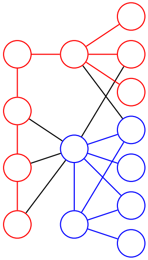
$(G, 6) \in$ 2-LC

The c-Load Coloring problem

c-Load Coloring

Input : graph $G = (V, E)$; integer $k \geq 0$ as parameter

Output : $\exists ?\ \varphi : V \to [c], \forall i \in [c], |E(\varphi^{-1}(i))| \geq k$



$(G, 7) \in$ 2-LC

First observations

- $(G, 0) \in c$-LC $(\forall c)$
- $(G, 1) \in c$-LC $\Leftrightarrow$ a maximum matching has at least c edges.
- $(G, k) \in 1$-LC $\Leftrightarrow$ G has at least k edges.

First observations

- $(G, 0) \in c$-LC $(\forall c)$
- $(G, 1) \in c$-LC $\Leftrightarrow$ a maximum matching has at least c edges.
- $(G, k) \in 1$-LC $\Leftrightarrow$ G has at least k edges.

$\Rightarrow$ we may assume $c, k > 1$.

The 2-LOAD COLORING case

Ahuja and al. 2007 : Introduction of 2-LOAD COLORING

- Application: broadcast WDM communication networks
- NP-Complete but Polynomial for trees.
- Constant approximation for some genus-related graph class.

The 2-Load Coloring case

Ahuja and al. 2007 : Introduction of 2-Load Coloring

- Application: broadcast WDM communication networks
- NP-Complete but Polynomial for trees.
- Constant approximation for some genus-related graph class.

Gutin and Jones, 2014 : Analysis with parameterized point of view

- Kernel with at most $7k$ vertices $\Rightarrow$ FPT simple-exponential
- Treewidth $tw(G)$ at most $2k$
- $O^*(2^{tw(G)}k^4)$-time dynamic programming algorithm.

Basics of Parameterized Complexity

Purpose: Refining complexity classes using multiple parameter analysis.

Fixed-Parameter Tractable (FPT)

Π is **FPT** if and only if Π has an $f(p)poly(n)$-time algorithm.

Basics of Parameterized Complexity

Purpose: Refining complexity classes using multiple parameter analysis.

Fixed-Parameter Tractable (FPT)

Π is **FPT** if and only if Π has an $f(p)poly(n)$-time algorithm.

Kernelization

Polynomial-time preprocessing: $(G, p) \to (G', p')$, the **kernel**.
Requirement:

- $(G, p) \in \Pi$ if and only if $(G', p') \in \Pi$
- $\exists g$, $\max\{|G'|, p'\} \leq g(p)$

Our contributions for c-Load Coloring

	$c = 2$	fixed $c \geq 2$
Complexity	NP-Complete FPT simple-exponential	NP
Kernel	$7k$ vertices $\frac{49}{2}k^2$ edges	n vertices m edges
Brute Force	$O^*(2^{7k})$	$O^*(c^n)$
FPT Algo	$O^*(2^{tw(G)}k^4)$ $treewidth \leq 2k$	
Approximation	genus-related	

Our contributions for c-Load Coloring

	$c = 2$	fixed $c \geq 2$
Complexity	NP-Complete FPT simple-exponential	NP-Complete
Kernel	$7k$ vertices $\frac{49}{2}k^2$ edges	n vertices m edges
Brute Force	$O^*(2^{7k})$	$O^*(c^n)$
FPT Algo	$O^*(2^{tw(G)}k^4)$ $treewidth \leq 2k$	
Approximation	genus-related	

Our contributions for c-LOAD COLORING

	$c = 2$	fixed $c \geq 2$
Complexity	NP-Complete FPT simple-exponential	NP-Complete FPT simple-exponential
Kernel	$7k$ vertices $\frac{49}{2}k^2$ edges	n vertices m edges
Brute Force	$O^*(2^{7k})$	$O^*(c^n)$
FPT Algo	$O^*(2^{tw(G)}k^4)$ $tree depth \leq 2k$	$O^*(c^{tw(G)}k^{2c})$ $treedepth < c(k+1)$
Approximation	genus-related	

Our contributions for c-Load Coloring

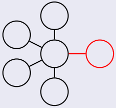
	$c = 2$	fixed $c \geq 2$
Complexity	NP-Complete FPT simple-exponential	NP-Complete FPT simple-exponential
Kernel	$4k$ vertices $8k^2$ edges	$2ck$ vertices $2c^2k^2$ edges
Brute Force	$O^*(2^{4k})$	$O^*(c^{2ck})$
FPT Algo	$O^*(2^{tw(G)}k^4)$ $treedepth \leq 2k$	$O^*(c^{tw(G)}k^{2c})$ $treedepth < c(k+1)$
Approximation	genus-related	

Our contributions for c-Load Coloring

	$c = 2$	fixed $c \geq 2$
Complexity	NP-Complete FPT simple-exponential	NP-Complete FPT simple-exponential
Kernel	$4k$ vertices $6(k + \sqrt{k})$ edges	$2ck$ vertices $6.25c^2k$ edges
Brute Force	$O^*(2^{4k})$	$O^*(c^{2ck})$
FPT Algo	$O^*(2^{tw(G)}k^4)$ $treedepth \leq 2k$	$O^*(c^{tw(G)}k^{2c})$ $treedepth < c(k+1)$
Approximation	$3 + \epsilon$	$6.25c + \epsilon$

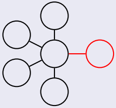
Observation on an optimal coloring φ

If uv is a pendant edge then $\varphi(u) = \varphi(v)$



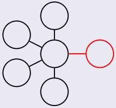
Observation on an optimal coloring φ

If uv is a pendant edge then $\varphi(u) = \varphi(v)$
$\Rightarrow$ Stars are monochromatic once colored by φ



Observation on an optimal coloring φ

If uv is a pendant edge then $\varphi(u) = \varphi(v)$
$\Rightarrow$ Stars are monochromatic once colored by φ
$\Rightarrow (G, k) \in$ 2-LC $\Leftrightarrow (G \uplus (c-2)K_{1,k}, k) \in c$-LC



Reduction from 2-LOAD COLORING to c-LOAD COLORING

Observation on deleting vertices

$$(G, k) \in c\text{-LC} \Rightarrow (G/W, k) \in (c - |W|)\text{-LC}$$

Observation on deleting vertices

$$(G, k) \in c\text{-LC} \Rightarrow (G/W, k) \in (c - |W|)\text{-LC}$$

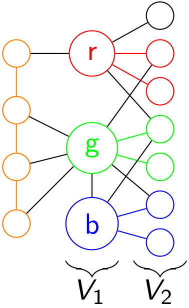
for all n, $(K_{c,n-c}, k) \notin (c+1)$-LC

at least a reduction rule must apply on unbalanced bipartite graphs.

Overload

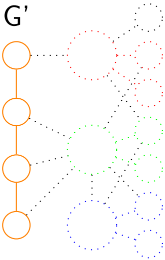
A pair (V_1, V_2) of disjoint vertex sets is an *overload* if

- V_2 is isolated in $G - V_1$
- There exists $|V_1|$ k-stars centered at V_1 with leaves in V_2



Reduction rule

$$(G, k) \text{ for } c\text{-LC} \rightarrow (G - (V_1 \cup V_2), k) \text{ for } (c - |V_1|)\text{-LC}$$



Linear Kernel for Star-Graph

Lemma 1

If G is a forest of stars, $\Delta < 2k$ and $n \geq 2ck$, then $(G, k) \in c$-LC

Linear Kernel for Star-Graph

Lemma 1

If G is a forest of stars, $\Delta < 2k$ and $n \geq 2ck$, then $(G, k) \in c$-LC

Intuition

Monochromatic stars with small degree $\approx$ small positive numbers

General Linear Kernel

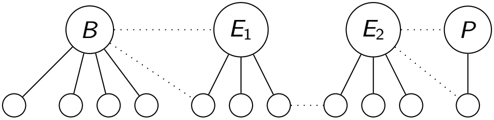
Theorem 1

There exists an $O((cn)^2)$-time algorithm which either finds:

- a small-degree star-cover (then $(G, k) \in c$-LC)
- an overload (V_1, V_2) s.t. $|V(G - (V_1 \cup V_2))| < 2(c - |V_1|)k$

General Linear Kernel (proof)

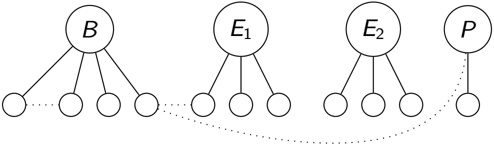
Isolated-vertex-free graphs have star-cover(s).



Case 1: BS Leaf isolated ⇒ reducible.

General Linear Kernel (proof)

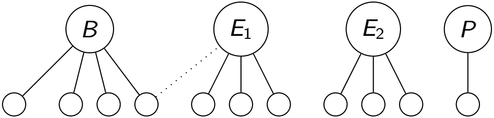
Isolated-vertex-free graphs have star-cover(s).



Case 2: Edge between BS leaf and not BS or ES center → improvement.

General Linear Kernel (proof)

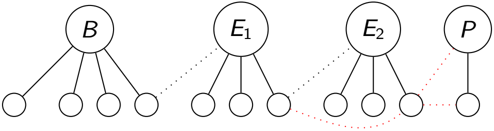
Isolated-vertex-free graphs have star-cover(s).



Question: Edge between BS leaf and BS or ES center?

General Linear Kernel (proof)

Isolated-vertex-free graphs have star-cover(s).



Case 3: Induce until Case 1 or 2.

More than only a linear-vertex Kernel

- Theorem 1 $\Rightarrow$ kernel with less than $2ck$ vertices.
- $(ck-1)$-matching = reduced No-Instance
  $\Rightarrow$ tight lower bound.

More than only a linear-vertex Kernel

- Theorem 1 $\Rightarrow$ kernel with less than $2ck$ vertices.
- $(ck-1)$-matching = reduced No-Instance
  $\Rightarrow$ tight lower bound.

But how many edges ?

More than only a linear-vertex Kernel

- Theorem 1 $\Rightarrow$ kernel with less than $2ck$ vertices.
- $(ck-1)$-matching = reduced No-Instance
  $\Rightarrow$ tight lower bound.

But how many edges ?

Theorem 2

The kernel has less than $2.25c^2k + 2cn \leq 6.25c^2k$ edges

Intuition

Bipartite graph with many edges can be split.

Some Generalizations

For pseudo-graph with multiplicity μ:

- Self-loop $\equiv$ pendant edge
- Still $2ck$-vertex kernel
- $\mu \leq k$ and $6.25c^2\mu k$-edge kernel

Some Generalizations

For pseudo-graph with multiplicity μ:

- Self-loop $\equiv$ pendant edge
- Still $2ck$-vertex kernel
- $\mu \leq k$ and $6.25c^2\mu k$-edge kernel

When we want k_i edges for color $i \in [c]$ ($k_{max} = \max k_i$) :

- $2ck_{max}$-vertex kernel
- $6.25c^2 k_{max}$-edge kernel
- treedepth $< \Sigma(k_i + 1)$

Some Generalizations

For pseudo-graph with multiplicity μ:

- Self-loop $\equiv$ pendant edge
- Still $2ck$-vertex kernel
- $\mu \leq k$ and $6.25c^2\mu k$-edge kernel

When we want k_i edges for color $i \in [c]$ $(k_{max} = \max k_i)$:

- $2ck_{max}$-vertex kernel
- $6.25c^2 k_{max}$-edge kernel
- treedepth $< \Sigma(k_i + 1)$

Still $O^*(c^{tw(G)} k_{max}^{2c})$-time Dynamic Programming algorithm.

Sub-exponential FPT Algorithm (1)

Recall that we have an $O^*(c^{tw(G)}k^{2c})$-time dynamic programming algorithm.

Theorem

There exists an $O^*(c^{O(g\sqrt{ck})})$-time dynamic programming algorithm solving c-Load Coloring, where $g = genus(G)$

Sub-exponential FPT Algorithm (1)

Recall that we have an $O^*(c^{tw(G)}k^{2c})$-time dynamic programming algorithm.

Theorem

There exists an $O^*(c^{O(g\sqrt{ck})})$-time dynamic programming algorithm solving c-LOAD COLORING, where $g = genus(G)$

Proof

Let G_r be an $(r \times r)$-grid. If $r \geq \lceil \sqrt{c(k+1)} \rceil$, $(G_r, k) \in c$-LC.
Since for any graph G, G_r is a minor of G for $r = O(\frac{t}{g})$, we have:

$$t \leq O(g\sqrt{ck}) \text{ or } (G, k) \in c\text{-LC}$$

Sub-exponential FPT Algorithm (2)

Recall that we have an $O^*(c^{tw(G)}k^{2c})$-time dynamic programming algorithm.

Theorem

There exists an $O^*(c^{c\sqrt{2k}})$-time dynamic programming algorithm solving c-LOAD COLORING restricted to chordal graphs.

Proof

$$\text{If } r \geq c\lceil\sqrt{2k}+1\rceil,\ (K_r, k) \in c\text{-LC}.$$

Since for any chordal graph G, K_t is a subgraph of G, we have:

$$t < c\lceil\sqrt{2k}+1\rceil \text{ or } (G, k) \in c\text{-LC}$$

Sub-exponential FPT Algorithm (3)

Theorem

For any fixed H, c-LOAD COLORING restricted to H-minor-free graph admits an $O^*(c^{O(\sqrt{k})})$-time algorithm.

Sub-exponential FPT Algorithm (3)

Theorem

For any fixed H, c-LOAD COLORING restricted to H-minor-free graph admits an $O^*(c^{O(\sqrt{k})})$-time algorithm.

Partial proof

The complement of c-LOAD COLORING is minor-bidimensional and admits an $O^*(c^{tw(G)})$-time dynamic programming algorithm.

New Open Questions

- Sharpen linear-edge kernel size?
  New reduction rules or optimality proof?
- Better use of treedepth bound?
- Sub-exponential algorithm in any case?
- Other Generalization? For instance, the $(c, k)H$ problem ?

New Open Questions

- Sharpen linear-edge kernel size?
  New reduction rules or optimality proof?
- Better use of treedepth bound?
- Sub-exponential algorithm in any case?
- Other Generalization? For instance, the $(c, k)H$ problem ?

Thank you for your attention !