Degrees and Gaps: Tight Complexity Results of General Factor Problems Parameterized by Treewidth and Cutwidth

ICALP 2021

Dániel Marx*, Govind Sankar†, and **Philipp Schepper***

* CISPA Helmholtz Center for Information Security
† Indian Institute of Technology Madras

14 July 2021

General Factor Problem

General Factor

Input: A simple graph $G = (V, E)$ and for each $v \in V$ a set $B_v \subseteq \mathbb{N}$.
Task: Check if there is a *solution* $S \subseteq E$, i.e. $\deg_S(v) \in B_v$ for all $v \in V$.

Generalizes Perfect Matching by setting $B_v = \{1\}$ for all $v \in V$.

General Factor Problem

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

General Factor

Input: A simple graph $G = (V, E)$ and for each $v \in V$ a set $B_v \subseteq \mathbb{N}$.
Task: Check if there is a *solution* $S \subseteq E$, i.e. $\deg_S(v) \in B_v$ for all $v \in V$.

Generalizes Perfect Matching by setting $B_v = \{1\}$ for all $v \in V$.

B-Factor

General Factor when $B_v = B$ for all v for some fixed, finite set $B \subseteq \mathbb{N}$.

What makes B-Factor easy or hard to solve: Size of the sets? Largest number?

General Factor Problem

General Factor

Input: A simple graph $G = (V, E)$ and for each $v \in V$ a set $B_v \subseteq \mathbb{N}$.
Task: Check if there is a *solution* $S \subseteq E$, i.e. $\deg_S(v) \in B_v$ for all $v \in V$.

Generalizes Perfect Matching by setting $B_v = \{1\}$ for all $v \in V$.

B-Factor

General Factor when $B_v = B$ for all v for some fixed, finite set $B \subseteq \mathbb{N}$.

What makes B-Factor easy or hard to solve: Size of the sets? Largest number?

Max-Gap

For finite $\emptyset \neq B \subseteq \mathbb{N}$, max-gap B is the largest d such that there is an a with $[a, a + d + 1] \cap B = \{a, a + d + 1\}$.
(At most d consecutive numbers are missing in B.)

Example: $\{2, 4, 8\}$ has gaps of size 1 and 3 = max-gap.

Known Results

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Cornuéjols '88)

B-Factor is solvable in polynomial time if max-gap $B \leq 1$.

Cornuéjols implicitly showed NP-hardness when max-gap > 1 but uses two lists ($\{0, 3\}$ and $\{1\}$).

Known Results

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Cornuéjols '88)

B-FACTOR is solvable in polynomial time if max-gap $B \leq 1$.

Cornuéjols implicitly showed NP-hardness when max-gap > 1 but uses two lists ($\{0, 3\}$ and $\{1\}$).

What about the graph structure? $\Rightarrow$ Treewidth, pathwidth, ...

Theorem (Arulselvan et al. '18)

B-FACTOR can be solved in time $(\max B + 1)^{3\text{tw}} n^{\mathcal{O}(1)}$, given a tree decomposition of width tw.

Open Questions: Is this optimal? Can we show lower bounds?

Our Results

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Upper Bound)

$(\max B + 1)^{\mathrm{tw}} n^{\mathcal{O}(1)}$ algorithm for counting solutions of a certain size given a tree decomposition of width tw of the graph.

Idea: Standard dynamic programming on tree decomposition combined with known convolution techniques for join nodes (van Rooij '20).

Our Results

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Theorem (Upper Bound)

$(\max B + 1)^{\text{tw}} n^{\mathcal{O}(1)}$ algorithm for counting solutions of a certain size given a tree decomposition of width tw of the graph.

Idea: Standard dynamic programming on tree decomposition combined with known convolution techniques for join nodes (van Rooij '20).

Theorem (Lower Bound)

Fix a $B \subseteq \mathbb{N}$ with max-gap $B > 1$ and $0 \notin B$. For all $\varepsilon > 0$, there is no $(\max B + 1 - \varepsilon)^{\text{tw}} n^{\mathcal{O}(1)}$ algorithm for B-Factor, given a tree decomposition of treewidth tw, unless SETH fails.

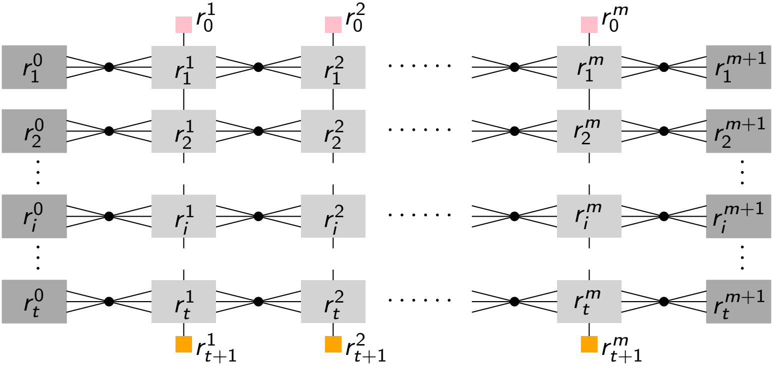
(Consequence of) Strong Exponential Time Hypothesis

There is no $\delta > 0$ such that CNF-SAT can be solved in time $(2 - \delta)^n$ on formulas with n variables.

Lower Bound: High Level Construction

Cannot use $n \times m$ grid: high treewidth $\rightarrow$ no tight lower bound.

- Group $\log(\max B + 1)$ variables: Grid with $n/\log(\max B + 1)$ rows.
- Encode partial assignments by selection of up to $\max B$ edges.
- Check at each crossing point if the partial assignment satisfies the clause.



Treewidth is $n/\log(\max B + 1) + \mathcal{O}(1) \Rightarrow (\max B + 1 - \varepsilon)^{\text{tw}} n^{\mathcal{O}(1)}$ lower bound.

Realizing Relations

It remains to model the check at the crossing points:

- Define relations with the correct behaviour.
- Replace these relations by graphs with the same behaviour (*realizations*).

Realizing Relations

It remains to model the check at the crossing points:

- Define relations with the correct behaviour.
- Replace these relations by graphs with the same behaviour (*realizations*).

Based on a result from Curticapean and Marx (2016) this boils down to:

- Forcing edges to be in the solution.
- Getting equality gadgets (either none or all incident edges are selected).

Realizing Relations

It remains to model the check at the crossing points:

- Define relations with the correct behaviour.
- Replace these relations by graphs with the same behaviour (*realizations*).

Based on a result from Curticapean and Marx (2016) this boils down to:

- Forcing edges to be in the solution.
- Getting equality gadgets (either none or all incident edges are selected).

Version	Decision	Maximization	Counting
Assumptions	max-gap $B > 1$ min $B > 0$	max-gap $B > 1$	B non-trivial
Equality	Use the gap and forced edges	Use the gap and forced edges	Use interpolation with weights to reduce to forced edges
Forced Edge	(min $B + 1$)-clique	High girth graphs to get a penalty	Interpolation

Conclusion

Upper Bounds for General Factor

We can count in time $(M+1)^{\text{tw}} n^{\mathcal{O}(1)}$ solutions of a certain size given a tree decomposition of width tw. (M is maximum over all sets B_v.)

Conclusion

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Upper Bounds for General Factor

We can count in time $(M+1)^{\mathrm{tw}} n^{\mathcal{O}(1)}$ solutions of a certain size given a tree decomposition of width tw. (M is maximum over all sets B_v.)

Lower Bounds Parameterizing by Treewidth

Fix a finite $B \subseteq \mathbb{N}$. For any $\varepsilon > 0$, there is no $(\max B + 1 - \varepsilon)^{\mathrm{tw}} n^{\mathcal{O}(1)}$ algorithm for the following problems, given a tree decomposition of width tw unless SETH (resp. #SETH) fails:

- B-Factor and Min-B-Factor if $0 \notin B$ and max-gap $B > 1$,
- Max-B-Factor if max-gap $B > 1$,
- #B-Factor if $B \neq \emptyset, \{0\}$.

Conclusion

Upper Bounds for General Factor

We can count in time $(M+1)^{\mathrm{tw}} n^{\mathcal{O}(1)}$ solutions of a certain size given a tree decomposition of width tw. (M is maximum over all sets B_v.)

Lower Bounds Parameterizing by Treewidth

Fix a finite $B \subseteq \mathbb{N}$. For any $\varepsilon > 0$, there is no $(\max B + 1 - \varepsilon)^{\mathrm{tw}} n^{\mathcal{O}(1)}$ algorithm for the following problems, given a tree decomposition of width tw unless SETH (resp. #SETH) fails:

- B-Factor and Min-B-Factor if $0 \notin B$ and max-gap $B > 1$,
- Max-B-Factor if max-gap $B > 1$,
- #B-Factor if $B \neq \emptyset, \{0\}$.

Bounds Parameterizing by Cutwidth

Analogous upper and lower bounds for a $2^{\mathrm{cutw}} n^{\mathcal{O}(1)}$ algorithm, when given a linear layout of width cutw.