The Closed Geodetic Game: algorithms and strategies

Antoine Dailly[1,2], Harmender Gahlawat[3], Zin Mar Myint[4]

P-GASE Workshop, October 28th, 2024

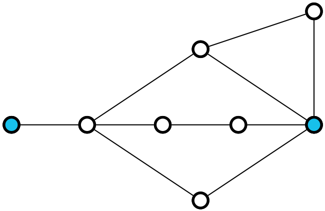
[1] LIMOS, Université Clermont-Auvergne, Clermont-Ferrand, France
[2] TSCF, INRAE, Clermont-Ferrand, France
[3] G-SCOP, Université Grenoble Alpes, France
[4] Indian Institute of Technology Dharwad, India

Funded by ANR GRALMECO, I-SITE CAP 20-25, Doctoral Fellowship in India for ASEAN DIA:2020-25.

Geodetic Sets

Geodetic closure [Harary & Nieminen, 1981]

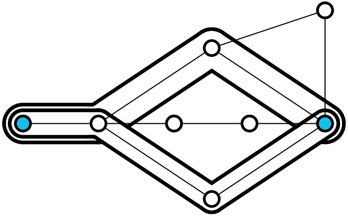
For a set S of vertices: the set of all vertices in shortest paths between vertices of S, denoted by $\langle S \rangle$.



Geodetic Sets

Geodetic closure [Harary & Nieminen, 1981]

For a set S of vertices: the set of all vertices in shortest paths between vertices of S, denoted by (S).



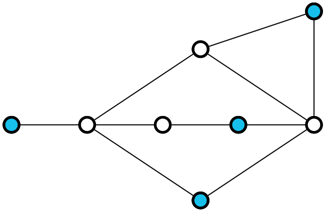
Geodetic Sets

Geodetic closure [Harary & Nieminen, 1981]

For a set S of vertices: the set of all vertices in shortest paths between vertices of S, denoted by (S).

Geodetic set [Buckley, Harary & Quintas, 1988]

A set S of vertices of graph $G(V, E)$ such that $(S) = V$.



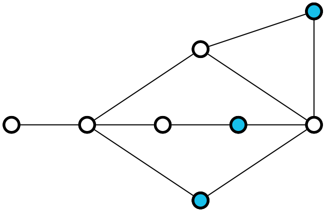
Geodetic Sets

Geodetic closure [Harary & Nieminen, 1981]

For a set S of vertices: the set of all vertices in shortest paths between vertices of S, denoted by (S).

Geodetic set [Buckley, Harary & Quintas, 1988]

A set S of vertices of graph $G(V, E)$ such that $(S) = V$.



- Many combinatorial and algorithmic results...

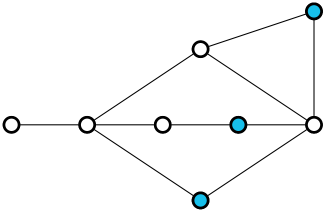
Geodetic Sets

Geodetic closure [Harary & Nieminen, 1981]

For a set S of vertices: the set of all vertices in shortest paths between vertices of S, denoted by (S).

Geodetic set [Buckley, Harary & Quintas, 1988]

A set S of vertices of graph $G(V, E)$ such that $(S) = V$.



- Many combinatorial and algorithmic results... which we will not care about in this talk!

Geodetic Games

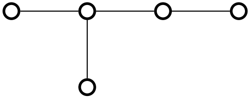
Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding vertices to S until it is geodetic.

Geodetic Games

Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding vertices to S until it is geodetic.

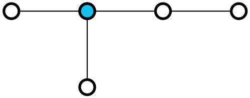


Let us play! (under *normal* convention)

Geodetic Games

Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding vertices to S until it is geodetic.

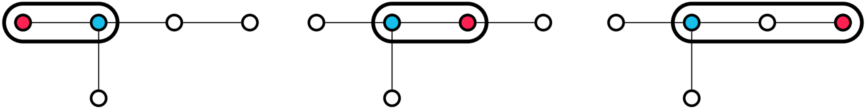


Let us play! (under *normal* convention)

Geodetic Games

Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding vertices to S until it is geodetic.

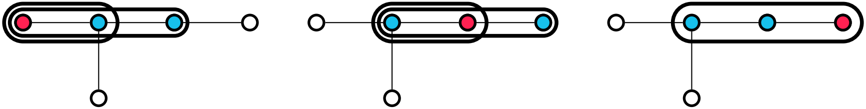


Let us play! (under *normal* convention)

Geodetic Games

Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding vertices to S until it is geodetic.

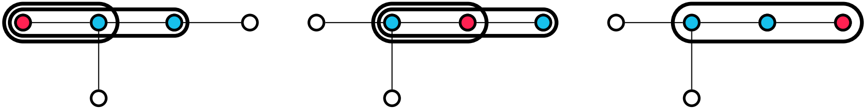


Let us play! (under *normal* convention)
Seems like I'm the best. 😎

Geodetic Games

Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding vertices to S until it is geodetic.



Let us play! (under *normal* convention)
Seems like I'm the best. 😎

- Complete graphs, cycles, complete bipartite graphs, n-cubes [Buckley & Harary, 1985]
- Generalized wheels [Nečásková, 1993]
- Complete multipartite graphs, hypercubes, graphs with a unique optimal geodetic set [Haynes, Henning & Tiller, 2003]

Geodetic Games II: Now Closed!

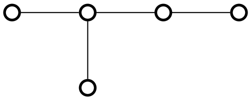
Closed Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding to S vertices **not in** $\langle \mathbf{S} \rangle$ until S is geodetic.

Geodetic Games II: Now Closed!

Closed Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding to S vertices **not in** $\langle \mathbf{S} \rangle$ until S is geodetic.

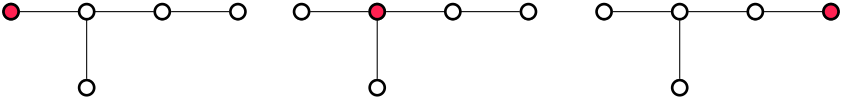


Let us play! (under *normal* convention) This time, you begin.

Geodetic Games II: Now Closed!

Closed Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding to S vertices **not in** $(\mathbf{S})$ until S is geodetic.

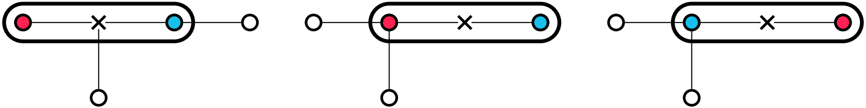


Let us play! (under *normal* convention) This time, you begin.

Geodetic Games II: Now Closed!

Closed Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding to S vertices **not in** $(\mathbf{S})$ until S is geodetic.

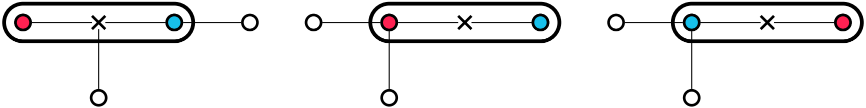


Let us play! (under *normal* convention) This time, you begin.
Well I'm still the best. 😎

Geodetic Games II: Now Closed!

Closed Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding to S vertices **not in** (**S**) until S is geodetic.



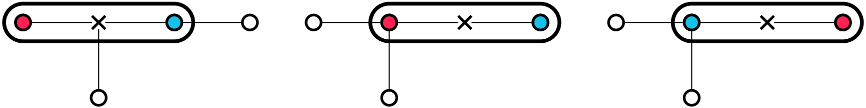
Let us play! (under *normal* convention) This time, you begin.
Well I'm still the best. 😎

- Complete graphs, cycles, complete bipartite graphs, n-cubes [Buckley & Harary, 1985]
- Linear-time algorithm for Grundy values of trees [Araujo *et al.*, 2024]

Geodetic Games II: Now Closed!

Closed Geodetic Game [Buckley & Harary, 1985]

Two players alternate adding to S vertices **not in** (**S**) until S is geodetic.



Let us play! (under *normal* convention) This time, you begin.
Well I'm still the best. 😎

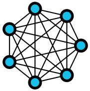
- Complete graphs, cycles, complete bipartite graphs, n-cubes [Buckley & Harary, 1985]
- Linear-time algorithm for Grundy values of trees [Araujo *et al.*, 2024]

→ We study the CLOSED GEODETIC GAME

First results: Grundy values

Some trivial ones

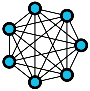
- $\mathcal{G}(K_n) = n \bmod 2$ (every vertex has to be selected)

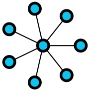


First results: Grundy values

Some trivial ones

- $\mathcal{G}(K_n) = n \bmod 2$ (every vertex has to be selected)
- $\mathcal{G}(K_{1,n}) = 1 - (n \bmod 2)$ (every vertex will be selected)

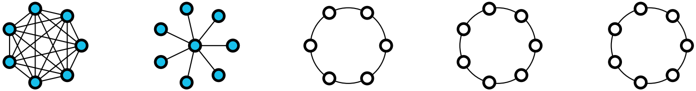




First results: Grundy values

Some trivial ones

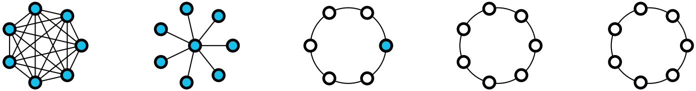
- $\mathcal{G}(K_n) = n \bmod 2$ (every vertex has to be selected)
- $\mathcal{G}(K_{1,n}) = 1 - (n \bmod 2)$ (every vertex will be selected)
- $\mathcal{G}(C_n) = n \bmod 2$ (symmetry strategy)



First results: Grundy values

Some trivial ones

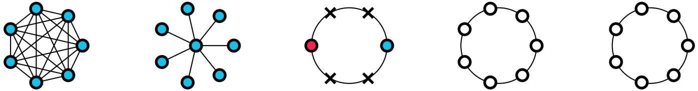
- $\mathcal{G}(K_n) = n \bmod 2$ (every vertex has to be selected)
- $\mathcal{G}(K_{1,n}) = 1 - (n \bmod 2)$ (every vertex will be selected)
- $\mathcal{G}(C_n) = n \bmod 2$ (symmetry strategy)



First results: Grundy values

Some trivial ones

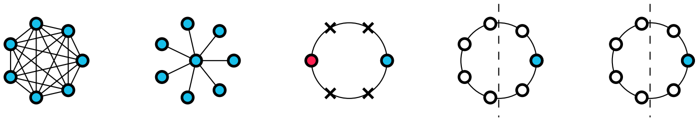
- $\mathcal{G}(K_n) = n \bmod 2$ (every vertex has to be selected)
- $\mathcal{G}(K_{1,n}) = 1 - (n \bmod 2)$ (every vertex will be selected)
- $\mathcal{G}(C_n) = n \bmod 2$ (symmetry strategy)



First results: Grundy values

Some trivial ones

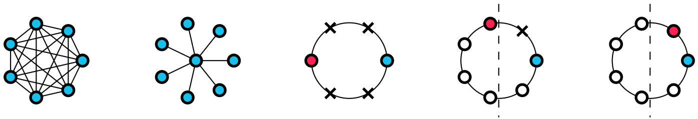
- $\mathcal{G}(K_n) = n \bmod 2$ (every vertex has to be selected)
- $\mathcal{G}(K_{1,n}) = 1 - (n \bmod 2)$ (every vertex will be selected)
- $\mathcal{G}(C_n) = n \bmod 2$ (symmetry strategy)



First results: Grundy values

Some trivial ones

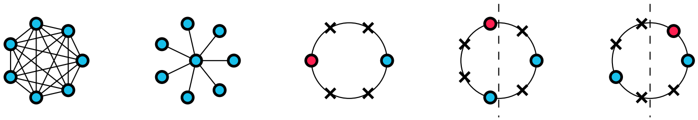
- $\mathcal{G}(K_n) = n \bmod 2$ (every vertex has to be selected)
- $\mathcal{G}(K_{1,n}) = 1 - (n \bmod 2)$ (every vertex will be selected)
- $\mathcal{G}(C_n) = n \bmod 2$ (symmetry strategy)



First results: Grundy values

Some trivial ones

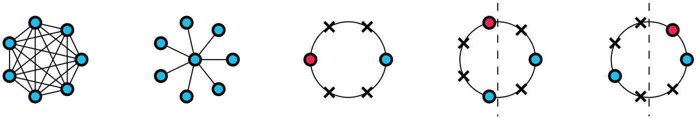
- $\mathcal{G}(K_n) = n \bmod 2$ (every vertex has to be selected)
- $\mathcal{G}(K_{1,n}) = 1 - (n \bmod 2)$ (every vertex will be selected)
- $\mathcal{G}(C_n) = n \bmod 2$ (symmetry strategy)



First results: Grundy values

Some trivial ones

- $\mathcal{G}(K_n) = n \bmod 2$ (every vertex has to be selected)
- $\mathcal{G}(K_{1,n}) = 1 - (n \bmod 2)$ (every vertex will be selected)
- $\mathcal{G}(C_n) = n \bmod 2$ (symmetry strategy)



Some less-trivial ones

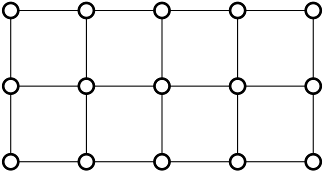
- $\mathcal{G}(P_n) = n \bmod 2$ (the value is expected, but the proof is nontrivial!)
- $\mathcal{G}(K_{m,n}) = 0$ if m and n have the same parity, and 2 otherwise

A fun result: grids

Proposition

A multidimensional grid has outcome $\mathcal{N}$ if and only if all its dimensions are odd.

Strategy



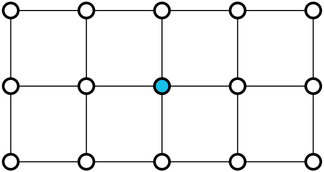
A fun result: grids

Proposition

A multidimensional grid has outcome $\mathcal{N}$ if and only if all its dimensions are odd.

Strategy

- First move: play in the middle vertex



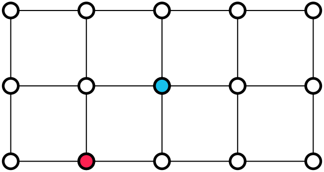
A fun result: grids

Proposition

A multidimensional grid has outcome $\mathcal{N}$ if and only if all its dimensions are odd.

Strategy

- First move: play in the middle vertex
- Afterwards:



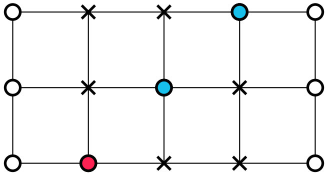
A fun result: grids

Proposition

A multidimensional grid has outcome $\mathcal{N}$ if and only if all its dimensions are odd.

Strategy

- First move: play in the middle vertex
- Afterwards: symmetry strategy!



Product results

Theorem [D., Gahlawat & Myint, 2024+]

For the Cartesian, tensor and strong products, the outcome of the product is $\mathcal{N}$ if and only if the outcomes of the two graphs are $\mathcal{N}$.

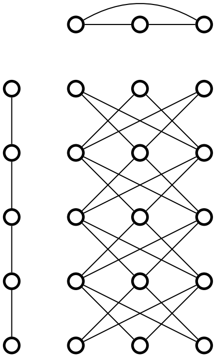
Product results

Theorem [D., Gahlawat & Myint, 2024+]

For the Cartesian, tensor and strong products, the outcome of the product is $\mathcal{N}$ if and only if the outcomes of the two graphs are $\mathcal{N}$.

Proof idea (for tensor product)

Assume G and H are $\mathcal{N}$.



Product results

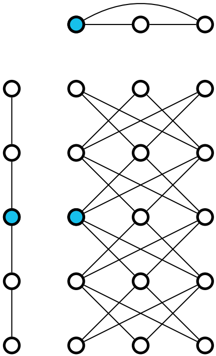
Theorem [D., Gahlawat & Myint, 2024+]

For the Cartesian, tensor and strong products, the outcome of the product is $\mathcal{N}$ if and only if the outcomes of the two graphs are $\mathcal{N}$.

Proof idea (for tensor product)

Assume G and H are $\mathcal{N}$.

- 1st move: apply the strategy



Product results

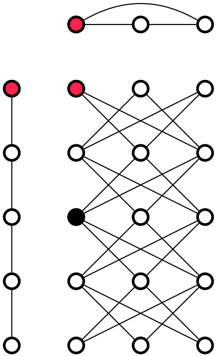
Theorem [D., Gahlawat & Myint, 2024+]

For the Cartesian, tensor and strong products, the outcome of the product is $\mathcal{N}$ if and only if the outcomes of the two graphs are $\mathcal{N}$.

Proof idea (for tensor product)

Assume G and H are $\mathcal{N}$.

- 1st move: apply the strategy
- Same row/column



Product results

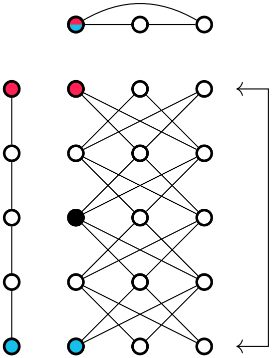
Theorem [D., Gahlawat & Myint, 2024+]

For the Cartesian, tensor and strong products, the outcome of the product is $\mathcal{N}$ if and only if the outcomes of the two graphs are $\mathcal{N}$.

Proof idea (for tensor product)

Assume G and H are $\mathcal{N}$.

- 1st move: apply the strategy
- Same row/column $\Rightarrow$ do the same, associate columns/rows



Product results

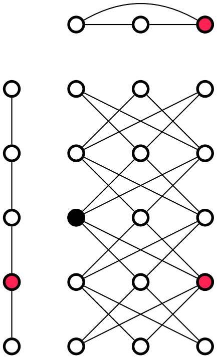
Theorem [D., Gahlawat & Myint, 2024+]

For the Cartesian, tensor and strong products, the outcome of the product is $\mathcal{N}$ if and only if the outcomes of the two graphs are $\mathcal{N}$.

Proof idea (for tensor product)

Assume G and H are $\mathcal{N}$.

- 1st move: apply the strategy
- Same row/column $\Rightarrow$ do the same, associate columns/rows
- Distinct move



Product results

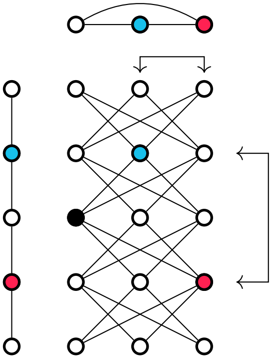
Theorem [D., Gahlawat & Myint, 2024+]

For the Cartesian, tensor and strong products, the outcome of the product is $\mathcal{N}$ if and only if the outcomes of the two graphs are $\mathcal{N}$.

Proof idea (for tensor product)

Assume G and H are $\mathcal{N}$.

- 1st move: apply the strategy
- Same row/column $\Rightarrow$ do the same, associate columns/rows
- Distinct move $\Rightarrow$ apply the strategy on both graphs, associate rows and columns



Product results

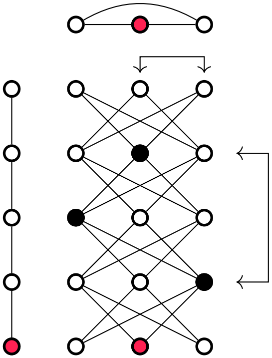
Theorem [D., Gahlawat & Myint, 2024+]

For the Cartesian, tensor and strong products, the outcome of the product is $\mathcal{N}$ if and only if the outcomes of the two graphs are $\mathcal{N}$.

Proof idea (for tensor product)

Assume G and H are $\mathcal{N}$.

- 1st move: apply the strategy
- Same row/column $\Rightarrow$ do the same, associate columns/rows
- Distinct move $\Rightarrow$ apply the strategy on both graphs, associate rows and columns
- Move on associated rows/columns



Product results

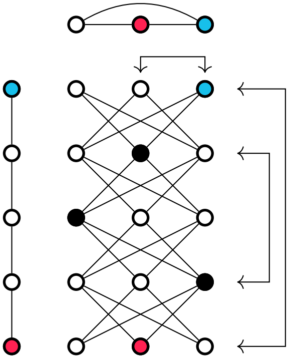
Theorem [D., Gahlawat & Myint, 2024+]

For the Cartesian, tensor and strong products, the outcome of the product is $\mathcal{N}$ if and only if the outcomes of the two graphs are $\mathcal{N}$.

Proof idea (for tensor product)

Assume G and H are $\mathcal{N}$.

- 1st move: apply the strategy
- Same row/column $\Rightarrow$ do the same, associate columns/rows
- Distinct move $\Rightarrow$ apply the strategy on both graphs, associate rows and columns
- Move on associated rows/columns $\Rightarrow$ Answer on associated, associate new rows/columns



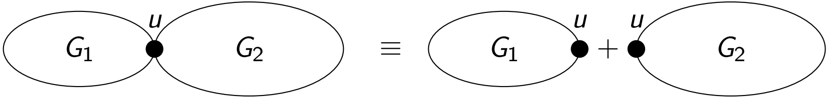
Algorithms for Grundy values

[Araujo *et al.*, 2024]'s algorithm for trees was based on the following:

Lemma

If u is an articulation point linking maximal components $G_1, \ldots, G_k$, then:

$$G, \{u\} \equiv (G_1, \{u\}) + \ldots + (G_k, \{u\}).$$



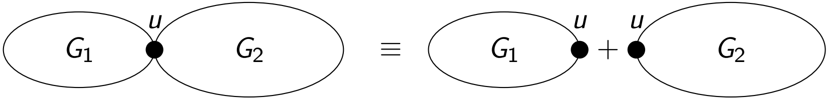
Algorithms for Grundy values

[Araujo *et al.*, 2024]'s algorithm for trees was based on the following:

Lemma

If u is an articulation point linking maximal components $G_1, \ldots, G_k$, then:

$$G, \{u\} \equiv (G_1, \{u\}) + \ldots + (G_k, \{u\}).$$



In a tree, every vertex is either a leaf or an articulation point $\Rightarrow$ Apply dynamic programing to compute the Grundy value

Algorithms for Grundy values: block graphs

Theorem [D., Gahlawat & Myint, 2024+]

There is a linear-time algorithm computing the Grundy values of block graphs.

Proof idea

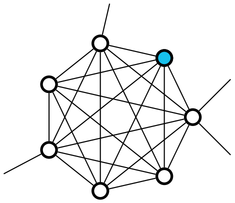
Algorithms for Grundy values: block graphs

Theorem [D., Gahlawat & Myint, 2024+]

There is a linear-time algorithm computing the Grundy values of block graphs.

Proof idea

- All non-articulation points moves on a given clique are equivalent



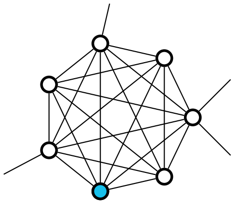
Algorithms for Grundy values: block graphs

Theorem [D., Gahlawat & Myint, 2024+]

There is a linear-time algorithm computing the Grundy values of block graphs.

Proof idea

- All non-articulation points moves on a given clique are equivalent



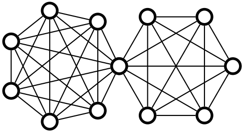
Algorithms for Grundy values: block graphs

Theorem [D., Gahlawat & Myint, 2024+]

There is a linear-time algorithm computing the Grundy values of block graphs.

Proof idea

- All non-articulation points moves on a given clique are equivalent
- Decompose after each move into subgraphs with at most one selected vertex



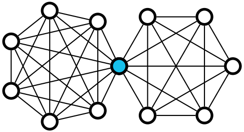
Algorithms for Grundy values: block graphs

Theorem [D., Gahlawat & Myint, 2024+]

There is a linear-time algorithm computing the Grundy values of block graphs.

Proof idea

- All non-articulation points moves on a given clique are equivalent
- Decompose after each move into subgraphs with at most one selected vertex



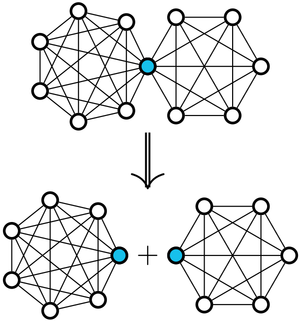
Algorithms for Grundy values: block graphs

Theorem [D., Gahlawat & Myint, 2024+]

There is a linear-time algorithm computing the Grundy values of block graphs.

Proof idea

- All non-articulation points moves on a given clique are equivalent
- Decompose after each move into subgraphs with at most one selected vertex



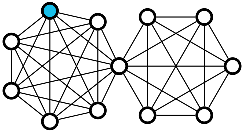
Algorithms for Grundy values: block graphs

Theorem [D., Gahlawat & Myint, 2024+]

There is a linear-time algorithm computing the Grundy values of block graphs.

Proof idea

- All non-articulation points moves on a given clique are equivalent
- Decompose after each move into subgraphs with at most one selected vertex



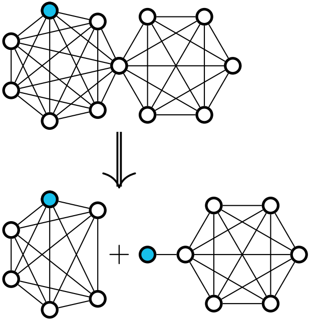
Algorithms for Grundy values: block graphs

Theorem [D., Gahlawat & Myint, 2024+]

There is a linear-time algorithm computing the Grundy values of block graphs.

Proof idea

- All non-articulation points moves on a given clique are equivalent
- Decompose after each move into subgraphs with at most one selected vertex



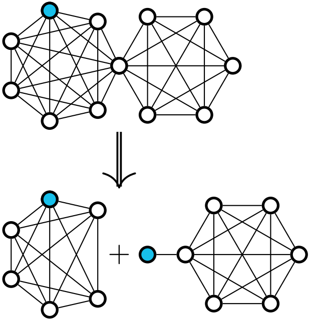
Algorithms for Grundy values: block graphs

Theorem [D., Gahlawat & Myint, 2024+]

There is a linear-time algorithm computing the Grundy values of block graphs.

Proof idea

- All non-articulation points moves on a given clique are equivalent
- Decompose after each move into subgraphs with at most one selected vertex
- Dynamic programing + storing intermediate values



Algorithms for Grundy values: cacti

Theorem [D., Gahlawat & Myint, 2024+]

There is a poly-time algorithm computing the Grundy values of cacti.

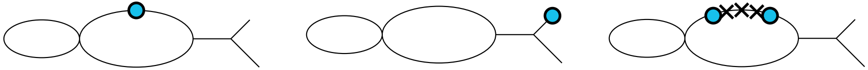
Algorithms for Grundy values: cacti

Theorem [D., Gahlawat & Myint, 2024+]

There is a poly-time algorithm computing the Grundy values of cacti.

Proof idea

- Three types of cacti with 1 or 2 selected vertices



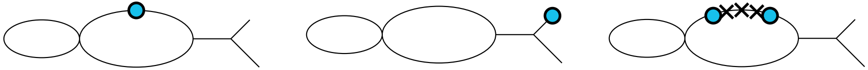
Algorithms for Grundy values: cacti

Theorem [D., Gahlawat & Myint, 2024+]

There is a poly-time algorithm computing the Grundy values of cacti.

Proof idea

- Three types of cacti with 1 or 2 selected vertices
- Each move in a type of cactus allows a decomposition into a sum of cacti of those types



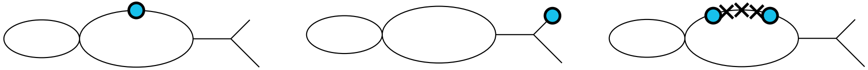
Algorithms for Grundy values: cacti

Theorem [D., Gahlawat & Myint, 2024+]

There is a poly-time algorithm computing the Grundy values of cacti.

Proof idea

- Three types of cacti with 1 or 2 selected vertices
- Each move in a type of cactus allows a decomposition into a sum of cacti of those types
- Dynamic programing + storing intermediate values



Final words

Our work

- Grundy values for structured classes
- Outcomes for products
- DP algorithms for Grundy values extending the ideas for trees

Final words

Our work

- Grundy values for structured classes
- Outcomes for products
- DP algorithms for Grundy values extending the ideas for trees

Future work

- Characterize graphs with parity Grundy values
- Other products
- Extend again the DP ideas to other decomposable graphs with strong geodetic structure

Final words

Our work

- Grundy values for structured classes
- Outcomes for products
- DP algorithms for Grundy values extending the ideas for trees

Future work

- Characterize graphs with parity Grundy values
- Other products
- Extend again the DP ideas to other decomposable graphs with strong geodetic structure