

# On Decompositions, Generation Methods, and related concepts in the theory of Matching Covered Graphs

On Decompositions, Generation Methods and related concepts in the theory of Matching Covered Graphs #38218

Student: Janmenjaya Panda

Roll No: ME20B087

Degree: B. Tech.

Guide: Nishad Kothari

janmenjaya commented 2 days ago

The objective of this issue is to... certain... tight cut... methods and related concepts in the theory of matching covered graphs.

This PR introduces new methods per... match... following:

1. The creation of a new class: `MatchingCoveredGraph`, in the module `sage.graphs.matching_covered_graph` and the implementation of the following functions:

- `maximal_barrier()`: Return the (unique) maximal barrier of the graph.
- `canonical_partition()`: Return the canonical partition of the graph.
- `tight_cut_decomposition()`: Return a maximal set of tight cuts whose shores wrt these cuts+ of the (matching covered) graph.
- `bricks_and_braces()`: Return the list of (underlying) bricks and braces of the (matching covered) graph.
- `number_of_bricks()`: Return the number of bricks of the graph.
- `number_of_braces()`: Return the number of braces of the graph.
- `number_of_petersen_bricks()`: Return the number of Petersen bricks of the graph. The graph is isomorphic to `graphs.petersen_graph()`.
- `is_brick()`: Check if the (matching covered) graph is a brick.
- `is_brace()`: Check if the (matching covered) graph is a brace.
- `is_removable_edge()`: Check if the edge of the (matching covered) graph is removable.



Indian Institute of Technology Madras

ent poly-time algorithms pert  
 al decision problems,  
 partition,  
 composition,  
 lations,  
 decomposition,  
 e generation methods, and  
 ts in the theory of matching  
 these available freely to studen  
 all across the world.

- ▶ the fundamental decision problems,
- ▶ the canonical partition,
- ▶ tight cut decomposition,
- ▶ dependency relations,
- ▶ (optimal) ear decomposition,
- ▶ brick and brace generation methods, and
- ▶ related concepts in the theory of matching covered graphs,

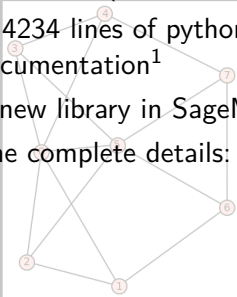
Springer

- ▶ Compilation of all existing algorithms available in the literature
- ▶ Devise of new/ equivalent definitions, theorems and derivations of efficient poly time algorithms from constructive/ existential proofs
- ▶ Implementing each of those in python/ cython with SageMath as the code base with the relevant documentations, examples and test cases
- ▶ Communicating with SageMath through Google Summer of Code (GSoC) 2024 to make all of these available publicly.

# Novelty and Results

```
1 G = Graph()
2 G.add_edges([(0, 1), (0, 2), (0, 3), (0, 4), (0, 5),
3             (1, 2), (1, 6), (2, 5), (3, 5), (3, 4),
4             (4, 7), (5, 6), (5, 7), (6, 7)])
5 M = Graph(G.matching())
6 show(G)
7
8 try:
9     answer = is_brick(G, M, matching_covered_check=True)
10    print('Is G a brick? {}'.format(answer))
11
12    print('error: {}'.format(error))
```

- ▶ 31 functional methods, 12 graph generator methods and 21 algorithms (10 existing algorithms, 11 formulated/ derived)
- ▶  $\approx 4234$  lines of python/ cython code and  $\approx 8807$  lines of documentation<sup>1</sup>
- ▶ A new library in SageMath: Matching Covered Graphs
- ▶ The complete details: [Link](#)



... Is G a brick? True

<sup>1</sup>An estimation

## Relevance

[illegible]