THE APPLICATION

PROJECT

Title	On Decompositions, Generation Methods and related concepts in the theory of Matching Covered Graphs
Broad subject	Matching theory, Graph theory, Algorithms, Theoretical computer science
Technical Details	The objective of this project is to implement efficient algorithms pertaining to 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, and to make all of these available freely to students, educators as well as researchers all across the world.

STUDENT

Name	Janmenjaya Panda
Roll No.	ME20B087
Mobile No.	6370642056
Degree	Bachelor of Technology
Program	Mechanical Engineering
Viva date	24th May 2024

GUIDE

Guide Name	Nishad Kothari
Designation	Assistant Professor
Joint - Guide (if any)	None
Designation	N/A

ABSTRACT

Matchings and perfect matchings have received considerable attention in graph theory as well as in other related domains (such as, but not limited to, algorithms and optimization). There still remain many open problems — such as Barnette's conjecture, Berge-Fulkerson conjecture, and so on — due to which it continues to remain an active area of research. For problems pertaining to perfect matchings, it is well-known that it suffices to solve them for matching covered graphs (that is, those connected graphs wherein each edge belongs to some perfect matching). The theory of matching covered graphs, despite its relatively recent emergence, presents an exciting landscape filled with captivating discoveries, elegant proofs, and unexpected applications. In the following paragraph, we briefly summarize some of the key developments in this field without going into the mathematical details.

Kotzig [8], in 1959, introduced the notion of canonical partition of a matching covered graph that uniquely partitions its vertex set into its maximal barriers. In 1987, László Lovász [11] established the uniqueness of the tight cut decomposition procedure; as per this, every matching covered graph may be uniquely decomposed into a list of special matching covered graphs called *"bricks"* (nonbipartite) and *"braces"* (bipartite). The key contribution of this landmark paper was to solve the Matching Lattice Problem. Lovász and Plummer, in 1975, introduced the well-known ear decomposition of matching covered graphs; see *"Matching Theory"* [10]. This notion of ear decomposition was further refined to that of an optimal ear decomposition by Carvalho, Lucchesi and Murty [6]. Furthermore, in their seminal paper, the same authors [2] introduced the dependency relationship in matching covered graphs that is closely tied with the ear decomposition theory. In 2001, McCuaig [13] established a generation method for all braces; analogously, Norine and Thomas [16], in 2007, established a generation method for all bricks. Both of these generation procedures may be viewed as a synthesis of the ear decomposition and tight cut decomposition theories, and have found applications in solving some of the major problems in matching theory — such as Polya's Permanent Problem [14]. All of these results — pertaining to decompositions, generation methods and related concepts — have played indispensable roles in the advancement of matching theory, and continue to do so.

It is worth noting that all of the notions discussed above are computable in poly-time. Despite this, there are no publicly available implementations. It is for this reason that researchers in this area are at a loss, and are required to implement parts of this theory by themselves. Currently, in SageMath, a few existing matching-theoretic algorithms for general graphs have been put within the module "Undirected graphs" — either under the submodule "Algorithmically hard stuff" (for instance: `matching_polynomial()`) or under "Leftovers" (for instance: `has_perfect_matching()`, `matching()`, `is_factor_critical()` and `perfect_matchings()`), whereas that concerning the bipartite graphs have been put within the module "Bipartite Graphs" (for instance: `matching()`, `matching_polynomial()` and `perfect_matchings()`). Ergo, we propose to implement efficient algorithms pertaining to the results and concepts discussed in the above paragraph in SageMath, and to make all of these available freely to students, educators as well as researchers all across the world. This proposal has been inspired by the book of Lucchesi and Murty — *"Perfect Matchings: a theory of matching covered graphs"* [12].

METHODOLOGY

All the graphs considered in this work are undirected and loopless. But, they might contain multiple edges. For graph theoretical notation and terminology, the main resources that are essentially followed, are — Graph Theory (2008, [1]) by Bondy and Murty. This work assumes that the reader has the basic knowledge in graph theory. The reader is requested to refer to the equivalent papers in case they require an in depth overview of the concerning concepts.

Fundamentals

For a graph $G := (V, E)$, a *matching* is any subset of the edge set E, say M, such that $|M \cap \partial(v)$[1]$| \leqslant 1$ for each vertex $v \in V$. Here, for a vertex v, the notation $\partial(v)$ denotes the set of edges incident at that vertex. We implement the Micali-Vazirani algorithm [15] for computing an (unweighted) maximum matching of a general graph even though it has a time complexity of $\mathcal{O}(\sqrt{|V|} \cdot |E|)$, which is the best known theoretical algorithm for this problem.

A matching is said to be *perfect matching* if $|M \cap \partial(v)$[2]$| = 1$ for each vertex $v \in V$. A graph is said to be *matchable* if it has a perfect matching. A matchable graph G is said to be *bicritical* if $G - u - v$ is matchable for every pair of distinct vertices u and v. Bicritcal graphs play a significant role in the theory of matching covered graphs. An $\mathcal{O}(|V| \cdot |E|)$ algorithm is implemented using the M-alternating tree search method [9].

A connected nontrivial graph wherein each edge participates in some perfect matching is called a *matching covered graph*. An $\mathcal{O}(|V| \cdot |E|)$ algorithm for non bipartite graph and an $\mathcal{O}(|E|)$ algorithm for bipartite graph is devised and implemented using the M-alternating tree search method [9] and using a theorem pertaining to bipartite matching covered graphs [12] respectively.

Canonical Partition

For a graph G, a subset B of the vertex set is a *barrier* if $|U| = o(G - B) - |B|$, where $|U| = |V(G)| - 2|M|$. Here $|M|$ denotes the cardinality of the maximum matching of G and $o(G - B)$ denotes the number of odd components in $G - B$. For a graph G, a barrier B is a maximal barrier if C is not a barrier for each C such that $B \subset C \subseteq V$. Kotzig [8], in 1959, showed that *the maximal barriers of a matching covered graph G partition its vertex set, and this partition is called its canonical partition.*

Henceforth, each vertex in a matching covered graph participates in a unique maximal barrier. Thus, the kotzig relation is an equivalence relation for a matching covered graph. We devise and implement a $\mathcal{O}(|E|)$ and $\mathcal{O}(|V| \cdot |E|)$ algorithm for computing the maximal barrier containing a particular vertex and computing the canonical partition of the matching covered graph respectively.

Tight cut decomposition

For a graph G and for a set $S \subseteq V(G)$, the notation $\partial(S)$ denotes the set of edges that have one end in S and the other end in $\overline{S} := V(G) - S$. A cut C of a matching covered graph G is called a *tight cut* if $|M \cap C| = 1$, for each perfect matching M of G. A matching covered graph free of tight cuts is called *a brace* if it is bipartite and *a brick* if it is nonbipartite. Given any matching covered graph G, we may apply to it a procedure, called a *tight cut decomposition* of G, which produces a list of bricks and braces.

Lovász [11] proved the following remarkable result on tight cut decomposition: *Any two distinct applications of the tight cut decomposition procedure to a matching covered graph G produce the same list of bricks and braces, up to multiple edges.* Without mentioning the definitions of laminar cuts, barrier cuts and 2-separation cuts, and Edmonds-Lovász-Pulleyblank cut [7], we state the following theorem that was conjectured by Carvalho, Lucchesi and Murty [3] and proved by Chen, Feng, Lu, Lucchesi, and Zhang [5]: *If C is a nontrivial tight cut in a matching covered graph G, then there exists a nontrivial ELP cut, that is, either a nontrivial barrier cut or a nontrivial 2-separation cut in G, that is laminar to C.*

We use the above theorem and some other results [12], to devise and implement an $\mathcal{O}(|V| \cdot |E|)$ algorithm to determine whether the graph is a brick, an $\mathcal{O}(|V| \cdot |E|)$ algorithm to check if the graph is a brace and an $\mathcal{O}(|V|^2 \cdot |E|)$ algorithm for the tight cut decomposition procedure.

Notable families of bricks and braces

We implement efficient generators for several notable families of bricks and braces, namely — Möbius ladder graph, biwheel graph, truncated biwheel graph, staircase graph and named matching covered, in particular — the Bicorn, the Tricorn, the Tretracorn graph, the Murty graph, the Meredith graph, the KohsTindell digraph, the Cubeplex graph and the Twinplex graph.

Dependency relation

Deletions and contractions of edges are two common inductive tools in graph theory. In 1999, through their landmark paper "Ear decompositions of matching covered graphs", Carvalho, Lucchesi and Murty [2] introduced the notation of dependency relation and removable classes in matching covered graphs, that are crucial for several significant results in the theory of matching covered graphs, for instance — computing an optimal ear decomposition of a matching covered graph (which we will see in the Section 'Ear decomposition').

An edge e of a matching covered graph G is removable if the graph $G-e$ is also matching covered and a pair $\{e, f\}$ of edges of a matching covered graph G is a removable doubleton if $G - e - f$ is matching covered, but neither $G - e$ nor $G - f$ is. Given a matching covered graph G, a perfect matching M and an edge e, we implement an $\mathcal{O}(\mathcal{V} \cdot |E|)$ algorithm if the graph is nonbipartite and $\mathcal{O}(|E|)$ algorithm if the graph is bipartite for deciding the particular edge is removable or not. And analogously, for a given pair of edges, we implement an $\mathcal{O}(|V| \cdot |E|)$ algorithm to determine if they constitute a removable doubleton. We also implement $\mathcal{O}(|E|^2 \cdot |V|)$ algorithm [3] for computing all removable edges and all removable doubletons for a given matching covered graph.

[3] $\mathcal{O}(|E|^2)$ algorithm for computing removable edges in bipartite graphs

Ear decomposition

The ear decomposition procedure has played a significant role in the theory of matching covered graph, as provides us with one of the excellent induction tools to investigate the properties of matching covered graphs. Here, we state the ear decomposition theorem for matching covered graph [12]: *Given any matching covered graph G there exists a sequence*

$$\mathcal{G} := (G_1 = G \supset G_2 \cdots \supset G_r = K_2)$$

of conformal matching covered subgraphs of G such that, for $1 \leqslant i \leqslant r-1$,

$$G_{i+1} = G_i - R_i, \text{ where } R_i \text{ is a removable ear of } G_i.$$

Carvalho, Lucchesi and Murty, [6], in 2002, proved a bound on the minimum number of ears required for an ear decomposition of a matching covered graph. The ear decomposition of a matching covered graph that includes minimum number of ears is called an *optimal ear decomposition*.

We implement an efficient $\mathcal{O}(|V|\cdot|E|)$ algorithm [4] for computing an arbitrary ear decomposition, an $\mathcal{O}(|E|)$ algorithm for computing the retract and an other algorithm [12] for computing an optimal ear decomposition of a matching covered graph.

Brick and brace generation

We state the following result of McCuaig [13] as stated in [12]:
Given any simple brace G of order six or more, there exists a sequence

$$G_1, G_2, \ldots, G_k$$

of simple braces such that:

1. *G_1 is either a biwheel, or a prism, or a Möbius ladder, and $G_k = G$, and*
2. *for $2 \leqslant i \leqslant k$, the graph G_i is obtained from G_{i-1} by an expansion operation.*

Analogously, we state the following result of Norine and Thomas [16] as stated in [12]: *Given any simple brick G, there exists a sequence*

$$G_1, G_2, \ldots, G_k$$

of simple bricks such that:

1. *G_1 is either a wheel, or a truncated biwheel, or a prism, or a Möbius ladder, or a staircase, or the Petersen graph, and $G_k = G$, and*
2. *for $2 \leqslant i \leqslant k$, the graph G_i is obtained from G_{i-1} by an expansion operation.*

Provided with a simple brick/ brace, we implement efficient methods/ algorithms to compute a brick/ brace generation sequence.

Summary and Results

- 31 functional methods, 12 graph generator methods and 21 algorithms [4]
  (10 existing algorithms, 11 formulated/ derived)
- ≈ 4234 lines of python/ cython code and ≈ 8807 lines of documentation[5]
- A new library in SageMath: Matching Covered Graphs [6]
- The complete details: Link

Signature of Guide Nishad

Signature of Co-guide N/A

Head of the department

REFERENCES

---

[1] Adrian Bondy and U.S.R. Murty. Graph Theory. Graduate Texts in Mathematics. Springer London, 2008.

[2] M. H. Carvalho, C. L. Lucchesi, and U. S. R. Murty. Ear decompositions of matching covered graphs. Combinatorica, 19:151–174, 1999.

[3] M. H. Carvalho, C. L. Lucchesi, and U. S. R. Murty. On tight cuts in matching covered graphs. Journal of Combinatorics, 9:163–184, 2018.

[4] M.H. Carvalho and J. Cheriyan. An o(v e) algorithm for ear decompositions of matching-covered graphs. ACM Trans. Algorithms, 1(1):324–337, 2005.

[5] G. Chen, X. Feng, F. Lu, C. L. Lucchesi, and L. Zhang. Laminar tight cuts in matching covered graphs. Journal of Combinatorial Theory, Series B, 150:177–194, 2021.

[6] M. H. de Carvalho, C. L. Lucchesi, and U. S. R. Murty. Optimal ear decompositions of matching covered graphs and bases for the matching lattice. Journal of Combinatorial Theory, Series B, 76(1):1–14, 2002.

[4] Note that there are several other methods that are implemented, which either serve as a dependent subroutine for these methods mentioned or some corollary to these.

[5] An estimation

[6] Note that it will be made public with the next (10.4) release of SageMath.

[7] J. Edmonds, L. Lovász, and W. R. Pulleyblank. Brick decompositions and the matching rank of graphs. Combinatorica, 2:247–274, 1982.

[8] A. Kotzig. Ein beitrag zur theorie der endlichen graphen i–ii–iii. Mat. Fyz. Casopis, 9:73–91, 136–159, and 10 (1960) 205–215, 1959.

[9] Dingjun Lou and Ning Zhong. A highly efficient algorithm to determine bicritical graphs. In Computing and Combinatorics, pages 349–356. Springer, 2001.

[10] L. Lovász and M. D. Plummer. Matching Theory. Number 29 in Annals of Discrete Mathematics. Elsevier Science, 1986.

[11] László Lovász. Matching structure and the matching lattice. Journal of Combinatorial Theory, Series B, 43(2):187–222, 1987.

[12] Cláudio L. Lucchesi and U.S.R. Murty. Perfect Matchings: A Theory of Matching Covered Graphs. Algorithms and Computation in Mathematics. Springer Cham, 1 edition, 25 March 2024.

[13] W. McCuaig. Brace generation. Journal of Graph Theory, 38:124–169, 2001.

[14] William McCuaig. Pólya's permanent problem. The Electronic Journal of Combinatorics, 11(1):R79, 2004.

[15] Silvio Micali and Vijay V. Vazirani. An $\mathcal{O}(\sqrt{|V|}\cdot|E|)$ algorithm for finding maximum matching in general graphs. In Proc. 21st IEEE FOCS, pages 17–27, 1980.

[16] S. Norine and R. Thomas. Generating bricks. Journal of Combinatorial Theory, Series B, 97:769–817, 2007.