Matching (graph theory)

From Wikipedia, the free encyclopedia

In the mathematical discipline of graph theory, a **matching** or **independent edge set** in a graph is a set of edges without common vertices. It may also be an entire graph consisting of edges without common vertices.

Contents

Covering-packing dualities

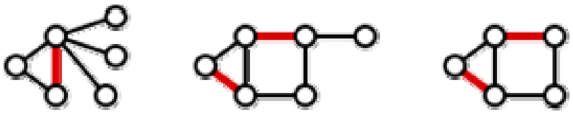
Covering problems	**Packing problems**
Minimum set cover	Maximum set packing
Minimum vertex cover	**Maximum matching**
Minimum edge cover	Maximum independent set

Definition

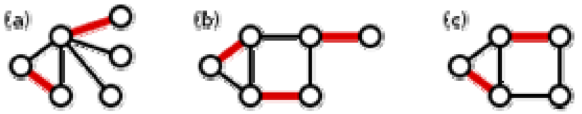
Given a graph *G* = (*V*,*E*), a **matching** *M* in *G* is a set of pairwise non-adjacent edges; that is, no two edges share a common vertex.

A vertex is **matched** (or **saturated**) if it is an endpoint of one of the edges in the matching. Otherwise the vertex is **unmatched**.

A **maximal matching** is a matching *M* of a graph *G* with the property that if any edge not in *M* is added to *M*, it is no longer a matching, that is, *M* is maximal if it is not a proper subset of any other matching in graph *G*. In other words, a matching *M* of a graph *G* is maximal if every edge in *G* has a non-empty intersection with at least one edge in *M*. The following figure shows examples of maximal matchings (red) in three graphs.



A **maximum matching** (also known as maximum-cardinality matching[1]) is a matching that contains the largest possible number of edges. There may be many maximum matchings. The **matching number** $\nu(G)$ of a graph G is the size of a maximum matching. Note that every maximum matching is maximal, but not every maximal matching is a maximum matching. The following figure shows examples of maximum matchings in the same three graphs.



A **perfect matching** (a.k.a. 1-factor) is a matching which matches all vertices of the graph. That is, every vertex of the graph is incident to exactly one edge of the matching. Figure (b) above is an example of a perfect matching. Every perfect matching is maximum and hence maximal. In some literature, the term **complete matching** is used. In the above figure, only part (b) shows a perfect matching. A perfect matching is also a minimum-size edge cover. Thus, $\nu(G) \leq \rho(G)$, that is, the size of a maximum matching is no larger than the size of a minimum edge cover.

A **near-perfect matching** is one in which exactly one vertex is unmatched. This can only occur when the graph has an odd number of vertices, and such a matching must be maximum. In the above figure, part (c) shows a near-perfect matching. If, for every vertex in a graph, there is a near-perfect matching that omits only that vertex, the graph is also called factor-critical.

Given a matching *M*,

- an **alternating path** is a path in which the edges belong alternatively to the matching and not to the matching.
- an **augmenting path** is an alternating path that starts from and ends on free (unmatched) vertices.

One can prove that a matching is maximum if and only if it does not have any augmenting path. (This result is sometimes called Berge's lemma.)

Properties

In any graph without isolated vertices, the sum of the matching number and the edge covering number equals the number of vertices.[2] If there is a perfect matching, then both the matching number and the edge cover number are $|V| / 2$.

If A and B are two maximal matchings, then $|A| \leq 2|B|$ and $|B| \leq 2|A|$. To see this, observe that each edge in $B \setminus A$ can be adjacent to at most two edges in $A \setminus B$ because A is a matching; moreover each edge in $A \setminus B$ is adjacent to an edge in $B \setminus A$ by maximality of B, hence

$$|A \setminus B| \leq 2|B \setminus A|.$$

Further we get that

$$|A| = |A \cap B| + |A \setminus B| \leq 2|B \cap A| + 2|B \setminus A| = 2|B|.$$

In particular, this shows that any maximal matching is a 2-approximation of a maximum matching and also a 2-approximation of a minimum maximal matching. This inequality is tight: for example, if G is a path with 3 edges and 4 nodes, the size of a minimum maximal matching is 1 and the size of a maximum matching is 2.

Matching polynomials

Main article: Matching polynomial

A generating function of the number of *k*-edge matchings in a graph is called a matching polynomial. Let *G* be a graph and m_k be the number of *k*-edge matchings. One matching polynomial of *G* is

$$\sum_{k\geq 0} m_k x^k.$$

Another definition gives the matching polynomial as

$$\sum_{k\geq 0} (-1)^k m_k x^{n-2k},$$

where *n* is the number of vertices in the graph. Each type has its uses; for more information see the article on matching polynomials.

Algorithms and computational complexity

Maximum matchings in bipartite graphs

Matching problems are often concerned with bipartite graphs. Finding a **maximum bipartite matching**[3] (often called a **maximum cardinality bipartite matching**) in a bipartite graph $G = (V = (X, Y), E)$ is perhaps the simplest problem. The **augmenting path algorithm** finds it by finding an augmenting path from each $x \in X$ to Y and adding it to the matching if it exists. As each path can be found in $O(E)$ time, the running time is $O(VE)$. This solution is equivalent to adding a *super source* s with edges to all vertices in X, and a *super sink* t with edges from all vertices in Y, and finding a maximal flow from s to t. All edges with flow from X to Y then constitute a maximum matching. An improvement over this is the Hopcroft-Karp algorithm, which runs in $O(\sqrt{V}E)$ time. Another approach is based on the fast matrix multiplication algorithm and gives $O(V^{2.376})$ complexity,[4] which is better in theory for sufficiently dense graphs, but in practice the algorithm is slower.

In a weighted bipartite graph, each edge has an associated value. A **maximum weighted bipartite matching**[3] is defined as a matching where the sum of the values of the edges in the matching have a maximal value. If the graph is not complete bipartite, missing edges are inserted with value zero. Finding such a matching is known as the assignment problem. It can be solved by using a modified shortest path search in the augmenting path algorithm. If the Bellman-Ford algorithm is used, the running time becomes $O(V^2E)$, or the edge cost can be shifted with a potential to achieve $O(V^2 \log V + VE)$ running time with the Dijkstra algorithm and Fibonacci heap.[5] The remarkable Hungarian algorithm solves the assignment problem and it was one of the beginnings of combinatorial optimization algorithms. The original approach of this algorithm needs $O(V^2E)$ running time, but it could be improved to $O(V^2 \log V + VE)$ time with extensive use of priority queues.

Maximum matchings

Main article: Edmonds's matching algorithm

There is a polynomial time algorithm to find a maximum matching or a maximum weight matching in a graph that is not bipartite; it is due to Jack Edmonds, is called the *paths, trees, and flowers* method or simply Edmonds's algorithm, and uses bidirected edges. A generalization of the same technique can also be used to find maximum independent sets in claw-free graphs. Edmonds' algorithm has subsequently been improved to run in time $O(\sqrt{V}E)$ time, matching the time for bipartite maximum matching.[6] Another algorithm by Mucha and Sankowski,[4] based on the fast matrix multiplication algorithm, gives $O(V^{2.376})$ complexity.

Maximal matchings

A maximal matching can be found with a simple greedy algorithm. A maximum matching is also a maximal matching, and hence it is possible to find a *largest* maximal matching in polynomial time. However, no polynomial-time algorithm is known for finding a **minimum maximal matching**, that is, a maximal matching that contains the *smallest* possible number of edges.

Note that a maximal matching with k edges is an edge dominating set with k edges. Conversely, if we are given a minimum edge dominating set with k edges, we can construct a maximal matching with k edges in polynomial time. Therefore the problem of finding a minimum maximal matching is essentially equal to the problem of finding a minimum edge dominating set.[7] Both of these two optimisation problems are known to be NP-hard; the decision versions of these problems are classical examples of NP-complete problems.[8] Both problems can be approximated within factor 2 in polynomial time: simply find an arbitrary maximal matching M.[9]

Counting problems

The problem of determining the number of perfect matchings in a given graph is #P-complete (see Permanent). However, a remarkable theorem of Kasteleyn states that the number of perfect matchings in a planar graph can be computed exactly in polynomial time via the FKT algorithm. There exists a fully polynomial time randomized approximation scheme for counting the number of bipartite matchings.[10]

The number of perfect matchings in a complete graph K_n (with n even) is given by the double factorial $(n-1)!!$.[11] The numbers of matchings in complete graphs, without constraining the matchings to be perfect, are given by the telephone numbers.[12]

For the problem of determining the total number of matchings in a given graph, see Hosoya index.

Finding all maximally-matchable edges

One of the basic problems in matching theory is to find in a given graph all edges that may be extended to a maximum matching in the graph. (Such edges are called **maximally-matchable** edges, or **allowed** edges.) The best deterministic algorithm for solving this problem in general graphs runs in time $O(VE)$.[13] There exists a randomized algorithm that solves this problem in time $\tilde{O}(V^{2.376})$.[14] In the case of bipartite graphs, it is possible to find a single maximum matching and then use it in order to find all maximally-matchable edges in linear time;[15] the resulting overall runtime is $O(V^{1/2}E)$ for general bipartite graphs and $O((V/\log V)^{1/2}E)$ for dense bipartite graphs with $E=\Theta(V^2)$. In cases where one of the maximum matchings is known upfront,[16] the overall runtime of the algorithm is $O(V+E)$.

Characterizations and Notes

König's theorem states that, in bipartite graphs, the maximum matching is equal in size to the minimum vertex cover. Via this result, the minimum vertex cover, maximum independent set, and maximum vertex biclique problems may be solved in polynomial time for bipartite graphs.

The marriage theorem (or Hall's Theorem) provides a characterization of bipartite graphs which have a perfect matching and the Tutte theorem provides a characterization for arbitrary graphs.

A perfect matching is a spanning 1-regular subgraph, a.k.a. a 1-factor. In general, a spanning *k*-regular subgraph is a *k*-factor.

Applications

A **Kekulé structure** of an aromatic compound consists of a perfect matching of its carbon skeleton, showing the locations of double bonds in the chemical structure. These structures are named after Friedrich August Kekulé von Stradonitz, who showed that benzene (in graph theoretical terms, a 6-vertex cycle) can be given such a structure.[17]

The Hosoya index is the number of non-empty matchings plus one; it is used in computational chemistry and mathematical chemistry investigations for organic compounds.

See also

- Edge coloring
- Vertex independent set
- Dulmage–Mendelsohn decomposition
- Stable matching
- Optimal matching
- Matching theory (macroeconomics)
- Skew-symmetric graph

References

1. ^ Alan Gibbons, Algorithmic Graph Theory, Cambridge University Press, 1985, Chapter 5.
2. ^ Gallai, Tibor (1959), "Über extreme Punkt- und Kantenmengen", *Ann. Univ. Sci. Budapest. Eötvös Sect. Math.* **2**: 133–138.
3. ^ ***a b*** West, Douglas Brent (1999), *Introduction to Graph Theory* (2nd ed.), Prentice Hall, Chapter 3, ISBN 0-13-014400-2
4. ^ ***a b*** Mucha, M.; Sankowski, P. (2004), "Maximum Matchings via Gaussian Elimination" (http://www.mimuw.edu.pl/~mucha/pub/mucha_sankowski_focs04.pdf) , *Proc. 45st IEEE Symp. Foundations of Computer Science*, pp. 248–255, http://www.mimuw.edu.pl/~mucha/pub/mucha_sankowski_focs04.pdf
5. ^ Fredman, M.; Tarjan, R. (1987), "Fibonacci heaps and their uses in improved network optimization algorithms" (http://dl.acm.org/citation.cfm?id=28874) , *Journal of the ACM (JACM)*, **34**, pp. 596–615, http://dl.acm.org/citation.cfm?id=28874
6. ^ Micali, S.; Vazirani, V. V. (1980), "An $O(\sqrt{|V|}\cdot|E|)$ algorithm for finding maximum matching in general graphs", *Proc. 21st IEEE Symp. Foundations of Computer Science*, pp. 17–27, doi:10.1109/SFCS.1980.12 (http://dx.doi.org/10.1109%2FSFCS.1980.12) .
7. ^ Yannakakis, Mihalis; Gavril, Fanica (1980), "Edge dominating sets in graphs", *SIAM Journal on Applied Mathematics* **38** (3): 364–372, doi:10.1137/0138030 (http://dx.doi.org/10.1137%2F0138030) .
8. ^ Garey, Michael R.; Johnson, David S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, ISBN 0-7167-1045-5. Edge dominating set (decision version) is

discussed under the dominating set problem, which is the problem GT2 in Appendix A1.1. Minimum maximal matching (decision version) is the problem GT10 in Appendix A1.1.

9. ^ Ausiello, Giorgio; Crescenzi, Pierluigi; Gambosi, Giorgio; Kann, Viggo; Marchetti-Spaccamela, Alberto; Protasi, Marco (2003), *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*, Springer. Minimum edge dominating set (optimisation version) is the problem GT3 in Appendix B (page 370). Minimum maximal matching (optimisation version) is the problem GT10 in Appendix B (page 374). See also Minimum Edge Dominating Set (http://www.nada.kth.se/~viggo/wwwcompendium/node13.html) and Minimum Maximal Matching (http://www.nada.kth.se/~viggo/wwwcompendium/node21.html) in the web compendium (http://www.nada.kth.se/~viggo/wwwcompendium/) .
10. ^ Bezáková, Ivona; Štefankovič, Daniel; Vazirani, Vijay V.; Vigoda, Eric (2008). "Accelerating Simulated Annealing for the Permanent and Combinatorial Counting Problems". *SIAM Journal on Computing* **37** (5): 1429–1454. doi:10.1137/050644033 (http://dx.doi.org/10.1137%2F050644033) .
11. ^ Callan, David (2009), *A combinatorial survey of identities for the double factorial*, arXiv:0906.1317 (http://arxiv.org/abs/0906.1317) .
12. ^ Tichy, Robert F.; Wagner, Stephan (2005), "Extremal problems for topological indices in combinatorial chemistry" (http://www.math.tugraz.at/fosp/pdfs/tugraz_main_0052.pdf) , *Journal of Computational Biology* **12** (7): 1004–1013, doi:10.1089/cmb.2005.12.1004 (http://dx.doi.org/10.1089%2Fcmb.2005.12.1004) , http://www.math.tugraz.at/fosp/pdfs/tugraz_main_0052.pdf.
13. ^ de Carvalho, Marcelo H.; Cheriyan, Joseph (2005), "An $O(VE)$ algorithm for ear decompositions of matching-covered graphs", *SODA*, pp. 415–423.
14. ^ Rabin, Michael O.; Vazirani, Vijay V. (1989), "Maximum matchings in general graphs through randomization", *J. of Algorithms* **10**: 557–567.
15. ^ Tassa, Tamir (2012), "Finding all maximally-matchable edges in a bipartite graph", *Theoretical Computer Science* **423**: 50–58, doi:10.1016/j.tcs.2011.12.071 (http://dx.doi.org/10.1016%2Fj.tcs.2011.12.071) .
16. ^ Gionis, Aris; Mazza, Arnon; Tassa, Tamir (2008), "*k*-Anonymization revisited", *International Conference on Data Engineering (ICDE)*, pp. 744–753.
17. ^ See, e.g., Trinajstić, Nenad; Klein, Douglas J.; Randić, Milan (1986), "On some solved and unsolved problems of chemical graph theory", *International Journal of Quantum Chemistry* **30** (S20): 699–742, doi:10.1002/qua.560300762 (http://dx.doi.org/10.1002%2Fqua.560300762) .

Further reading

1. László Lovász; M. D. Plummer (1986), *Matching Theory*, North-Holland, ISBN 0-444-87916-1
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein (2001), *Introduction to Algorithms* (second ed.), MIT Press and McGraw-Hill, Chapter 26, pp. 643–700, ISBN 0-262-53196-8
3. András Frank (2004). *On Kuhn's Hungarian Method – A tribute from Hungary* (http://www.cs.elte.hu/egres/tr/egres-04-14.pdf) . Egerváry Research Group. http://www.cs.elte.hu/egres/tr/egres-04-14.pdf.
4. Michael L. Fredman and Robert E. Tarjan (1987), "Fibonacci heaps and their uses in improved network optimization algorithms", *Journal of the ACM* (ACM Press) **34** (3): 595–615, doi:10.1145/28869.28874 (http://dx.doi.org/10.1145%2F28869.28874) .
5. S. J. Cyvin and Ivan Gutman (1988), *Kekule Structures in Benzenoid Hydrocarbons*, Springer-Verlag

External links

- A graph library with Hopcroft-Karp and Push-Relabel based maximum cardinality matching implementation (http://lemon.cs.elte.hu/)

Retrieved from "http://en.wikipedia.org/w/index.php?title=Matching_(graph_theory)&oldid=545253217"

Categories: Matching | Combinatorial optimization | Polynomial-time problems