

Fall 10-1-1975

A Note on Degree-Constrained Star Subgraphs of Bipartite Graphs ; CU-CS-083-75

Harold N. Gabow
University of Colorado Boulder

Follow this and additional works at: http://scholar.colorado.edu/csci_techreports

Recommended Citation

Gabow, Harold N., "A Note on Degree-Constrained Star Subgraphs of Bipartite Graphs ; CU-CS-083-75" (1975). *Computer Science Technical Reports*. 81.
http://scholar.colorado.edu/csci_techreports/81

This Technical Report is brought to you for free and open access by Computer Science at CU Scholar. It has been accepted for inclusion in Computer Science Technical Reports by an authorized administrator of CU Scholar. For more information, please contact cuscholaradmin@colorado.edu.

A NOTE ON DEGREE--CONSTRAINED
STAR SUBGRAPHS OF BIPARTITE GRAPHS

by

Harold N. Gabow
Department of Computer Science
University of Colorado
Boulder, Colorado 80309

CU-CS-083-75

October, 1975
Revised September, 1976

Abstract

Let f be a positive-valued function on the vertices of a graph G . An f -star subgraph F consists of vertex-disjoint stars, where each vertex v meets at most $f(v)$ edges of F ; a maximum f -star subgraph contains the greatest number of edges possible. Let G be bipartite, with vertex sets S, T ; let $f(v) = 1$ for all $v \in S$. Then there is a maximum f -star subgraph that contains a maximum matching. Let G have a perfect matching, let $f(v) = 1$ for $v \in S$, $f(v) \geq 2$ for $v \in T$. Then an f -star subgraph covering all vertices of S can be found in $O(E \log V)$ time. Also, a collection of vertex-disjoint paths of length 1 or 2 covering all vertices can be found in $O(E \log V)$ time.

Key words and phrases: bipartite graph, star, degree-constrained subgraph, matching.

This work was partially supported by the National Science Foundation under Grant GJ-09972.

Much work has been done on finding degree-constrained subgraphs of bipartite graphs $[T,U]$, and the special case of matchings [HK]. In this note we investigate a notion of intermediate generality, "f-star subgraphs". We indicate how these subgraphs relate to matchings, and how, in some cases, they can be constructed efficiently.

We first define some terms and notation. Standard terms not defined here are in [H]. Throughout this note, we limit ourselves to a bipartite graph $G = (S,T)$; here S and T are the two sets that partition the vertices. The degree of a vertex v is denoted $d(v)$, or $d(v,G)$ if the graph is not apparent.

Let f be a function assigning a positive integer to each vertex. A degree-constrained f-subgraph is a subgraph F , such that each vertex v meets at most $f(v)$ edges of F , i.e., $d(v,F) \leq f(v)$. (Note we assume $f(v) \geq 1$. This is without loss of generality, since if $f(v) = 0$, then v can be deleted from G .) If $d(v,F) = f(v)$, v is saturated; if $d(v,F) < f(v)$, v is covered. A maximum f-subgraph contains the greatest number of edges possible. A matching is an f -subgraph where $f(v) = 1$ for each vertex v . A perfect matching covers every vertex.

Let F be an f -subgraph. A path P is alternating (with respect to F) if it is simple (i.e., no vertex is repeated), and its edges are alternately in F and in $G-F$. P is augmenting if it is alternating, the first and last edges are in $G-F$, and the first and last vertices are unsaturated. If P is augmenting, the edges in $F \oplus P = F \cup P - F \cap P$ form an f -subgraph containing one more edge than F . If F is not a maximum f -subgraph, an augmenting path exists. This follows from network flow theory [FF].

A star is a tree consisting of a root and a number of sons. We define an f-star subgraph as an f-subgraph F that consists of vertex-disjoint stars. A maximum f-star subgraph contains the greatest number of edges possible. If $G=(S,T)$, then F is rooted in T if the root of each star is in T . An f-star subgraph rooted in T is equivalent to a g -subgraph, where $g(v) = 1$ for $v \in S$ and $g(v) = f(v)$ for $v \in T$. So an f-star subgraph rooted in T is a special case of a degree-constrained subgraph. In particular, the results on augmenting paths apply.

We begin by showing a relationship between f-star subgraphs and matchings.

Theorem 1: Let F be a maximum f-star subgraph rooted in T , covering the greatest number of vertices possible. Then a maximum matching M on F is a maximum matching on G .

Proof: First observe M is constructed by choosing one edge from each star of F . Now suppose M is not maximum, so it has an augmenting path P . We derive a contradiction below.

Let the first and last vertices of P be $v \in S$ and $w \in T$. Vertex w is not covered by M , so it is not covered by F . This implies v meets some edge $vu \in F-M$. For otherwise, v is not saturated by F (we assume $f(v) \geq 1$ by definition), and P is augmenting for F , a contradiction.

Since vertex $u \in T$ is covered by F , it is covered by M . Thus subgraph $F-vu$ covers only one less vertex than F . The edges in $(F-vu) \oplus P$ form an f-star subgraph with as many edges as F , covering one more vertex (w). This contradicts the definition of F . QED

Not every maximum matching on G can be obtained, as in Theorem 1, from a maximum f-star subgraph. This can be seen by letting G be the path $(v_1, v_2, v_3, v_4, v_5)$, $T = \{v_2, v_4\}$, and $f(v_i) = 1$ for $i \neq 2$, $f(v_2) = 2$.

A maximum f -star subgraph rooted in T can be constructed by Dinic's algorithm for network flows [D]. The run time is $O(V^{1/2}E+V)$. This can be shown by methods similar to those in [HK,T]. Now we refine this analysis, improving it for graphs with perfect matchings.

Theorem 2: Let F be an f -star subgraph rooted in T , where $f(v) \geq 2$ for all vertices $v \in T$. Let m be the number of edges in a maximum matching on G . If $|F| < m$, there is an augmenting path of length at most $2 \lfloor \log(|F|+2) \rfloor - 1$ [†].

Proof: Choose a maximum matching M so $|M \cap F|$ is as large as possible. There is a vertex $s \in S$, covered by M but not by F . Let C be the connected component of $M \oplus F$ that contains s .

Consider a simple path in C that starts at s , $(s=v_0, v_1, \dots, v_n)$. It is easy to see vertices $v_{2i} \in S, v_{2i+1} \in T$, and edges $v_{2i}v_{2i+1} \in M, v_{2i+1}v_{2i+2} \in F$, for all i where these vertices and edges exist. So the path is alternating. It also follows there is a unique simple path from any vertex in C to s . So C is a connected acyclic graph, i.e., C is a tree.

Make s the root of tree C . Let $2\ell+1$ be the length of the shortest augmenting path for F . We show by induction that for $0 \leq i \leq \ell$, level $2i$ of tree C contains at least 2^i vertices of S , and level $2i+1$ contains at least 2^i vertices of T .

The base step $i=0$ is trivial. For the inductive step, suppose the above assertion is true for some $i < \ell$. Let v be a vertex on level $2i+1$. Vertex v is saturated by F ; for otherwise, the tree path from

[†] Throughout this note, all logarithms are to the base 2.

v to s is augmenting for F , with length less than $2\ell+1$, a contradiction. So v has at least $f(v) \geq 2$ sons on level $2i+2$. Thus there are at least 2^{i+1} vertices on level $2i+2$.

Now let w be a vertex on level $2i+2$. Vertex w is covered by M ; for otherwise, if P is the tree path from w to s , $M \oplus P$ is a matching of the same cardinality as M , containing more edges of F , a contradiction. So w has a son on level $2i+3$. Thus there are at least 2^{i+1} vertices on level $2i+3$. This completes the induction.

Levels $2i$, $i=1, \dots, \ell$, of C contain $\sum_{i=1}^{\ell} 2^i = 2^{\ell+1} - 2$ vertices of S . Each of these vertices is covered by a distinct edge in F . Thus $2^{\ell+1} - 2 \leq |F|$, from which the desired inequality follows. QED

Dinic's algorithm constructs a maximum f -star subgraph by repeatedly augmenting, using shortest length augmenting paths. All augmenting paths of length ℓ are found together in time $O(E)$ [D]. So we have the following time bounds. As above, let m be the number of edges in a maximum matching, and assume $f(v) \geq 2$ for all vertices $v \in T$.

Corollary 1: Dinic's algorithm constructs an f -star subgraph rooted in T with at least m edges, in $O(E \log V+V)$ time.

Corollary 2: If a maximum matching covers all the vertices of S , then Dinic's algorithm constructs a maximum f -star subgraph rooted in T in $O(E \log V+V)$ time.

Sometimes the hypothesis of Corollary 2 can be verified without actually finding a maximum matching. For example, if $\min \{d(v) | v \in S\} \geq \max \{d(w) | w \in T\}$, then Hall's Theorem [FF] guarantees a maximum matching covers all vertices of S .

As an application of Corollary 2, consider a communications network composed of transmitter stations and relay stations. A message

goes from a transmitter to a relay and then to its destination. Each relay r handles up to $f(v) \geq 2$ messages simultaneously. The network operates as follows. When one or more previously idle transmitters get new messages, a graph is constructed. It contains a vertex for each previously active or newly active transmitter (S) and for each relay (T), and an edge for each possible transmission path. Then a maximum f -star subgraph rooted in T is found, and used to route messages. Note this subgraph may switch a previously active transmitter from one relay to another; presumably this does not harm transmission of the message. Note also if the possible transmission paths between all transmitters and all relays satisfy the degree constraint given above for Hall's Theorem, then all graphs constructed do too. So the f -star subgraph is found in the time bound of Corollary 2.

Now we consider general f -star subgraphs of a bipartite graph with a perfect matching. A maximum matching is an f -star subgraph that covers all vertices; it can be found in $O(E^{\frac{1}{2}}V+V)$ time [HK]. The following algorithm improves this time bound when $f \geq 2$.

procedure C;

comment This algorithm finds an f -star subgraph covering all vertices of a bipartite graph $G=(S,T)$ with a perfect matching;

begin

1. construct H , a maximum f -star subgraph rooted in T ; comment use Dinic's algorithm;
2. construct M , a maximum matching on H ; comment choose an edge from each star of H ;
3. construct F , a maximum f -star subgraph rooted in S , covering all

vertices covered by M ; comment use Dinic's algorithm, starting with M as the initial subgraph;

4. for each vertex $v \in T$ do
 - begin
 5. let A be the star of all edges $vw \in H$, where w is not covered by F ;
 6. if A has at least one edge then
 - begin
 7. let B be the star in F containing v ;
 8. if B has at least two edges then
 - delete the edge of B containing v from F ;
 9. add A to F ;
 - end;
 - end;
- end;

Theorem 3: Let G be a bipartite graph with a perfect matching; let $f(v) \geq 2$ for all vertices v . Algorithm C constructs an f -star subgraph F covering all vertices, in $O(E \log V)$ time.

Proof: First we show F is constructed correctly. Line 1 constructs a maximum f -star subgraph H rooted in T . H covers all vertices in S , since G has a perfect matching. Similarly, line 3 constructs an f -star subgraph F covering all vertices in T . F also covers all vertices covered by M , assuming M is the initial subgraph in Dinic's algorithm. For Dinic's algorithm works by repeatedly augmenting M , and in an augment, no covered vertex gets uncovered. So after line 3, F covers all vertices except possibly vertices $w \in S$ not covered by M . Now we show the loop in lines 4-9 covers these vertices.

Vertex w is in a star of H . So at some point, lines 4-5 choose v and A so edge wv is in star A . Vertex v is covered by exactly one edge, vu , of a star B in F . If B contains no edge besides vu , then line 9 adds A to F ; this covers w , without violating any constraints. (In particular, v meets at most $f(v)$ edges, since v meets fewer than $f(v)$ edges of A). If B has at least two edges, then line 8 deletes vu , and line 9 adds A ; this covers w , without uncovering u or violating any constraints. So in both cases, F is modified correctly to cover w . After the loop, F covers all vertices. So algorithm \mathcal{C} is correct.

Now consider the run time. Lines 1 and 3 use $O(E \log V)$ time, by Corollary 2. Lines 2, 4-9 use $O(E+V)=O(E)$ time, with the appropriate choice of data structures. So the total time is $O(E \log V)$. QED

In the special case where $f(v)=2$ for all vertices v , algorithm \mathcal{C} covers all vertices by a collection of vertex-disjoint paths of length 1 or 2.

ACKNOWLEDGEMENT

The author thanks the referees for their suggestions, and in particular, for observing that Theorem 1 was originally stated incorrectly.

REFERENCES

- [D] Dinic, E. A., "Algorithm for solution of a problem of maximum flow in a network with power estimation", Sov. Math. Dokl. 11, 5 (1970), 1277-1280.
- [FF] Ford, L. R., and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, N.J., 1962.
- [H] Harary, F., Graph Theory, Addison-Wesley, Reading, Mass., 1969.
- [HK] Hopcroft, J. E., and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs", SIAM J. Comput. 2, 4 (Dec. 1973), 225-231.
- [T] Tarjan, R. E., "Testing graph connectivity", Proc. 6th Annual ACM Symp. on Theory of Computing, Seattle, Washington, 1974, 185-193.
- [U] Urquhart, R. J., "Degree-constrained subgraphs of linear graphs", Ph.D. Diss., University of Michigan, 1967.