Maximum Vertex-Weighted Matching in Strongly Chordal Graphs*

Manoel B. Campêlo Neto† Sulamita Klein‡

Abstract

Given a graph $G = (V, E)$ and a real weight for each vertex of G, the vertex-weight of a matching is defined to be the sum of the weights of the vertices covered by the matching. In this paper we present a linear time algorithm for finding a maximum vertex-weighted matching in a strongly chordal graph, given a strong elimination ordering. The algorithm can be specialized to find a maximum cardinality matching, yielding an algorithm similar to one proposed earlier by Dahlhaus and Karpinsky. The technique does not seem to apply to the case of general edge-weighted matchings.

1 Introduction

The purpose of this paper is to present a linear time algorithm for finding a maximum vertex-weighted matching in a strongly chordal graph, given a strong elimination ordering.

We consider simple finite undirected graphs $G = (V, E)$. The number of vertices of G is denoted by n and the number of edges by m.

A subset M of edges of G ($M \subseteq E$) is a **matching** if no two edges in M are adjacent, i.e. share a vertex. If an edge $e = vw$ belongs to a matching M, we say that M covers the vertices v and w.

Given a real weight for each vertex of G, the **vertex-weight of a matching** M is defined to be the sum of the weights of the vertices covered by M [6]. Therefore, a **maximum vertex-weighted matching** is one which has the maximum vertex-weight among all possible matchings of G. If all the weights are equal, a maximum vertex-weighted matching is simply a maximum cardinality matching, i.e., a **maximum matching**.

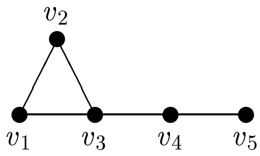
Let $v_1, v_2, \ldots, v_n$ be an ordering of the vertices of $G = (V, E)$. Then $e_1, e_2, \ldots, e_m$ is the **lexicographic ordering of the edges** (LOE) of G, with respect to the given ordering $v_1, v_2, \ldots, v_n$ of vertices, if for any two edges $e_i = v_{i_1}v_{i_2}$, with $i_1 < i_2$, and $e_j = v_{j_1}v_{j_2}$, with $j_1 < j_2$, we have $i < j$ if and only if $i_1 < j_1$, or $i_1 = j_1$ and $i_2 < j_2$ (see figure 1).

The lexicographic ordering of edges can be extended to any ordered sets of edges: Given two ordered edge-sets $E' = (e_{i_1}, e_{i_2}, \ldots, e_{i_r})$, with $i_1 < i_2 < \ldots < i_r$, and $E'' = (e_{j_1}, e_{j_2}, \ldots, e_{j_s})$, with

*This research was partially supported by CNPq and PICDT-CAPES

†Dep. Estatística e Matemática Aplicada, Univ. Federal do Ceará, Fortaleza-CE, 60455-760, Brasil; e-mail: mcampelo@ufc.br

‡Dep. Ciência da Computação/I.M. and COPPE/Sistemas, Univ. Federal do Rio de Janeiro, RJ, 21945-970, Brasil; e-mail: sula@cos.ufrj.br



LOE: $v_1v_2, v_1v_3, v_2v_3, v_3v_4, v_4v_5$

Figure 1: LOE - a lexicographical ordering of the edges of a strongly chordal graph, given a strong elimination ordering of its vertices.

$j_1 < j_2 < \ldots < j_s$, we say that E' is lexicographically less than E'' (and we write $E' \prec E''$) if there is an integer $t > 0$, such that $i_k = j_k$, for $k = 1, \ldots, t-1$, and $i_t < j_t$.

For $v_i, v_j \in V$, we denote $v_i \sim v_j$ when $i = j$ or $v_iv_j \in E$ (v_i adjacent to v_j). Similarly, for $e_i, e_j \in E$, the notation $e_i \sim e_j$ means $i = j$ or e_i adjacent to e_j. If there is not ambiguity, we prefer to write $i \sim j$, instead of $v_i \sim v_j$ or $e_i \sim e_j$. We write conjunctions of the adjacency relation $\sim$ with order relations such as $<, \leq, >, \geq$, so that, for instance, $i \stackrel{\sim}{>} j$ means $i \sim j$ and $i > j$.

An ordering $v_1, v_2, \ldots, v_n$ of V is a **strong elimination ordering** (SEO) of G if, for all i, j, k and l, with $i \leq j$, $k \leq l$ and $j \sim k \sim i \sim l$, we have $j \sim l$. A graph is **strongly chordal** if and only if it admits a SEO [2].

We will give a linear time algorithm for finding a maximum vertex-weighted matching in a strongly chordal graph, given a SEO. We begin in section 2 by presenting an integer linear programming formulation of our problem, and discussing it in the context of complementary slackness. In section 3 we describe an algorithm that solves our problem in this formulation. There we also prove its correctness and analyze its complexity. Section 4 specializes the technique to the particular problem of maximum matching, previously studied in [1].

2 ILP Formulation

Throughout this paper, we assume that $G = (V, E)$ is a strongly chordal graph, with a LOE $e_1, e_2, \ldots, e_m$, with respect to a SEO $v_1, v_2, \ldots, v_n$. Let w_i be the real weight associated to the vertex v_i. To each edge $e_i = v_{i_1}v_{i_2}$ we assign the weight $c_i = w_{i_1} + w_{i_2}$. Then the vertex-weight of a matching M is $w(M) = \sum_{e_i \in M} c_i$. We also define, for each i, the set $E_i = \{e_j | j \stackrel{\sim}{\geq} i\}$.

Lemma 1 *Let $M \subseteq E$. If $|M \cap E_i| \leq 1$, for $i = 1, \ldots, m$, then M is a matching.*

Proof: Suppose M is not a matching. Then there are two distinct adjacent edges $e_i, e_j \in M$. Without loss of generality, we can assume $j > i$. Therefore $j \stackrel{\sim}{>} i$, that is, $e_j \in E_i$. Then $\{e_i, e_j\} \subset M \cap E_i$ and, therefore, $|M \cap E_i| > 1$. ∎

Lemma 2 *Let M' be a matching. Then there is a matching M, with $w(M) = w(M')$, such that $|M \cap E_i| \leq 1$, for $i = 1, \ldots, m$.*

Proof: Let M' be a matching. If M' satisfies the desired condition, take $M = M'$. Otherwise, let i be such that $|M' \cap E_i| > 1$. Then there are two distinct edges $e_j, e_k \in M'$, such that $j \stackrel{\sim}{>} i$ and

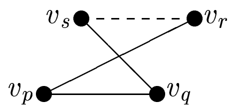


Figure 2:

$k \stackrel{\sim}{>} i$. Without loss of generality, suppose $j < k$ and $e_i = v_p v_q$, with $p < q$. Then, $e_j = v_p v_r$ and $e_k = v_q v_s$, where $q < r$ and $p < s$, because $e_1, e_2, \ldots, e_m$ is a LOE and $j < k$ (see figure 2).

Since $v_1, v_2, \ldots, v_n$ is a SEO and $r \sim p \sim q \sim s$, then $r \sim s$. Moreover $v_r v_s \in E$, since $r \neq s$. Define a new matching $M'' = M' - e_j - e_k + e_i + v_r v_s$, which covers the same vertices and has the same weight. Considering M' and M'' ordered subsets of E, note that $M'' \prec M'$, because $i < j < k$.

Therefore, if $|M' \cap E_i| > 1$, for some i, it is possible to generate a new matching M'' lexicografically less than M' with the same vertex-weight. Repeating the process, from M'', by a finite number of times, we can construct a matching M lexicographically minimum, with $w(M) = w(M')$. Then for this M, we have $|M \cap E_i| \leq 1$, for $i = 1, \ldots, m$. ■

Corollary 1 *Every strongly chordal graph has a maximum vertex-weighted matching $M \subseteq E$ which satisfies $|M \cap E_i| \leq 1$, for all $i = 1, \ldots, m$.*

Now we are ready to formulate our ILP model. To every edge e_i assign a variable $x_i \in \{0,1\}$. Then a solution of the following integer linear program gives a maximum vertex-weighted matching in a strongly chordal graph.

$$
\begin{array}{lll}
(P) & \text{maximize} & \displaystyle\sum_{i=1}^{m} c_i x_i \\
& \text{subject to} & \\
& & \begin{array}{rcll} \displaystyle\sum_{i \stackrel{\sim}{\geq} j} x_i & \leq & 1 & j = 1, \ldots, m \\ x_i & \in & \{0,1\} & i = 1, \ldots, m \end{array}
\end{array}
$$

Indeed, $M = \{e_i | x_i = 1\}$ is a maximum vertex-weighted matching. Note that $w(M) = \sum_{i=1}^{m} c_i x_i$ and $|M \cap E_j| = \sum_{i \stackrel{\sim}{\geq} j} x_i$. Hence the constraints of (P) force M to be a matching, and the objective function maximizes its vertex-weight.

The linear relaxation $(\overline{P})$ of (P) is obtained when the integrality constraints $x_i \in \{0,1\}$ are replaced by $x_i \geq 0$. The dual problem of $(\overline{P})$ is

$$
\begin{array}{lll}
(\overline{D}) & \text{minimize} & \displaystyle\sum_{j=1}^{m} z_j \\
& \text{subject to} & \\
& & \begin{array}{rcll} \displaystyle\sum_{j \stackrel{\sim}{\leq} i} z_j & \geq & c_i & i = 1, \ldots, m \\ z_j & \geq & 0 & j = 1, \ldots, m \end{array}
\end{array}
$$

3 The Algorithm

A subset $S \subseteq V$ is an independent set in G if any two vertices of S are not adjacent in G. With this concept in mind, we can say that a matching is an independent set of edges.

We intend to show that a procedure similar to one proposed by Farber [2] for finding a maximum weight independent set in chordal graphs can be used to find a maximum vertex-weighted matching in strongly chordal graphs.

The strategy to solve the problem consists of finding a pair of solutions that verify the constraints of $(\overline{P})$ and $(\overline{D})$, and also satisfy the complementary slackness conditions. Moreover, the solution of $(\overline{P})$ will be integer and, therefore, an optimum for (P).

The algorithm first generates a greedy solution of $(\overline{D})$, and then uses it to find a solution for $(\overline{P})$. In the first stage (dealing with the dual), the edges are scanned in the order $e_1, e_2, \ldots, e_m$ of the LOE. When the edge e_i is scanned, we assign to the variable z_i just the minimum necessary to bring that dual constraint associated to e_i into feasibility (possibly zero). In the second (primal) stage, we scan the edges in inverse order $e_m, e_{m-1}, \ldots, e_1$. When e_j is scanned, we assure that if $z_j > 0$ then the primal constraint associated with e_j should be satisfied with equality. If necessary, we let $x_j = 1$ to assure this.

Algorithm 1 .
Input: A strongly chordal graph G, a SEO $v_1, v_2, \ldots, v_n$ of G, and vertex-weights $w_1, w_2, \ldots, w_n$.
Output: A maximum vertex-weighted matching $M = \{e_i : x_i = 1\}$ given by a solution to (P).

Construct the LOE $e_1, e_2, \ldots, e_m$ of G with respect to $v_1, v_2, \ldots, v_n$
(Dual Stage *)*
  For $i = 1, 2, \ldots, m$ do
    let $e_i = v_{i_1} v_{i_2}$ and $c_i = w_{i_1} + w_{i_2}$
    $z_i = \max\{0, c_i - \sum_{j \tilde{<} i} z_j\}$
(Primal Stage *)*
  For $j = m, m-1, \ldots, 1$ do
    If $\sum_{i \tilde{>} j} x_i = 0$ and $z_j > 0$
    then $x_j = 1$
    else $x_j = 0$

The validity of the algorithm is established considering the duality between $(\overline{P})$ and $(\overline{D})$.

Theorem 1 *The algorithm 1 finds a maximum vertex-weighted matching.*

Proof: We clearly have $x_i \in \{0, 1\}$. Thus it remains to prove that $x = [x_i]$ and $z = [z_i]$ satisfy the constraints of $(\overline{P})$ and $(\overline{D})$ respectively, and that they satisfy the complementary slackness conditions.

- Feasibility of dual solution: For each i, by the choice of z_i, we have $z_i \geq 0$ and $c_i - \sum_{j \overset{\sim}{\leq} i} z_j \leq 0$.
- Feasibility of primal solution: If $x_i = x_j = 1$, clearly e_i is not adjacent to e_j. Since $x_i \in \{0, 1\}$, it follows that $\sum_{i \overset{\sim}{\geq} j} x_i \leq \sum_{i \sim j} x_i \leq 1$, for all $j = 1, 2, \ldots, m$.
- Complementary slackness: If $x_i > 0$, then $z_i > 0$, whence $c_i = \sum_{j \overset{\sim}{\leq} i} z_j$. On the other hand, when $z_j > 0$ then $\sum_{i \overset{\sim}{\geq} j} x_i \geq 1$ and equality follows from primal feasibility.

■

Theorem 2 *Given a SEO, the algorithm 1 halts after $O(n+m)$ operations.*

Proof: During the construction of the LOE, for each v_k with degree $d(v_k)$, we consider other $O(d(v_k))$ vertices. So this step takes $O(\sum_{k=1}^{n} d(v_k)) = O(m)$ time. The dual stage can be implemented by defining, for each v_k, a counter $L(k)$, initially zero, which stores (at the begining of each iteration $i = 1, 2, \ldots, m$) the sum of all variables z_j, with $j \overset{\sim}{<} i$ restricted to the edges incident to v_k. So, when we consider $e_i = v_{i_1} v_{i_2}$, we let $z_i = \max\{0, c_i - L(i_1) - L(i_2)\}$ and we add z_i to $L(i_1)$ and $L(i_2)$. Thus, the initialization is done in $O(n)$ time and the iterative step in $O(m)$. The primal stage is also done in linear time, provided we keep, for each v_k, a flag $S(k) \in \{0, 1\}$, such that $S(k) = 1$ if and only if v_k is covered. Initially we set $S(k) = 0$ for each v_k, which takes $O(n)$ operations. Thus for each $e_i = v_{i_1} v_{i_2}$, if $S(v_{i_1}) = S(v_{i_2}) = 0$, we let $x_i = S(v_{i_1}) = S(v_{i_2}) = 1$; otherwise, we let $x_i = 0$. It follows that the iterative step is done with $O(m)$ operations. ■

We note that a SEO of a strongly chordal graph can be found in $O(m \log n)$ [4, 7] time. Thus, the maximum weight-vertex matching can be solved in $O(m \log n)$ time in this class of graphs.

4 Maximum Matching

We now observe that when all the weights assigned to the vertices of the graph are equal, the problem considered is simply the problem of finding a maximum matching.

For simplicity, let us assume that $w_i = 0.5$, for $i = 1, 2, \ldots, n$, which implies $c_j = 1$, for $j = 1, 2, \ldots, m$. In this case, we note that the next result follows from the algorithm 1.

Lemma 3 *For all i, $x_i = z_i$.*

Proof: Since all the weights are integer values, each z_i is a non-negative integer. And, if $x_i = 1$, then $z_i > 0$. Hence $x_i \leq z_i$. Also, as we have $\sum_{i=1}^{n} (x_i - z_i) = 0$, it follows that $x_i = z_i$, for all i.

■

By the above considerations we can make the following simplifications in the algorithm 1. Since $z_i = \max\{0, 1 - \sum_{j \overset{\sim}{<} i} z_j\}$, we have $z_i > 0$, that is, $z_i = 1$, when $z_j = 0$, for all $j \overset{\sim}{<} i$. Moreover, the dual stage can be expressed in terms of the primal variables.

After such modifications we obtain an algorithm 2, similar to one proposed by Dahlhaus & Karpinsky [1], that has a structure identical to the procedure described by Gavril [3] for finding a maximum cardinality independent set in a chordal graph.

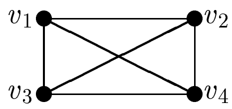


Figure 3:

Algorithm 2 .

Input: A strongly chordal graph G and a SEO $v_1, v_2, \ldots, v_n$ of G.

Output: A maximum matching M of G determined by those edges e_i with $x_i = 1$.

Construct a LOE $e_1, e_2, \ldots, e_m$ of G with respect to $v_1, v_2, \ldots, v_n$

For $i = 1, 2, \ldots, m$ do

If $\sum_{j \stackrel{\sim}{<} i} x_j = 0$

then $x_i = 1$

else $x_i = 0$

Unfortunately, the algorithm cannot be directly generalized to solve the maximum matching problem with arbitrary weights assigned to the edges, because the lemma 2 is no longer valid in this case. For example, suppose that the weigths of the edges in the figure 3 are proportional to their lenghts in the drawing. Then, the single maximum matching is formed by the edges v_1v_4 e v_2v_3, so that the statement of the corollary 1 is not satisfied by the edge v_1v_2, now considering edge-weights.

5 Conclusion

It is known that the maximum matching problem, in an arbitrary graph, can be solved in $O(m\sqrt{n})$ time [5]. However, it has been shown that this complexity can be reduced to $O(m \log n)$ time in strongly chordal graphs. Moreover, when a SEO is known a priori, the problem turns out to be linear.

Acknowledgements: We would like to express our thanks to Pavol Hell for his careful reading and helpful comments.

References

[1] E. Dahlhaus & M. Karpinsky. On the computational complexity of matching in chordal and strongly chordal graphs. Tech. Report TR-94-043, University of Bonn (1994).

[2] M. Farber. *Applications of linear programming duality to problems involving independence and domination.* Phd thesis, Rutgers University (1982).

[3] F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph. *SIAM J. Comput.* 1 (1972) 180-187.

[4] A. Lubiw. Doubly lexical orderings of matrices. *SIAM J. Comput.* 16 (1987) 854-879.

[5] S. Micali & V. V. Vazirani. An $o(\sqrt{|V|}|E|)$ algortihm for finding maximum matching in general graphs. *Proc. 21th Annual IEEE Symp. on the Foundations of Computer Science* (1980) 17-27.

[6] K. Mulmuley, U. V. Vazirani & V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica* 7 (1987) 105-113.

[7] R. Paige & R.E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.* 16 (1987) 973-989.