*Matemática Contemporânea, Vol 39, 69-76*
https://doi.org/10.21711/231766362010/rmc398


# Decomposition by maxclique separators

**Márcia R. Cerioli*** **Hugo Nobrega†** **Petrucio Viana‡**

**Abstract**

We provide a minimal counterexample to the correctness of an algorithm proposed by R. Tarjan for decomposing a graph by maximal clique separators. We also suggest a modification to that algorithm which not only corrects it but also retains its $O(nm)$ time complexity.

## 1 Introduction

Procedures for decomposing graphs into smaller pieces often play a central role in graph theory. Particularly, a type of graph decomposition which has found many interesting applications is that of decomposition by *clique separators*. In [5], R. Tarjan proposed an $O(nm)$ algorithm that decomposes a graph by clique separators, and showed how these decompositions can be used to efficiently solve many classical problems such as vertex coloring, maximum independent set, among others, in some graph classes.

Tarjan added a note at the end of his paper proposing a simple modification of his algorithm to find a decomposition by maximal clique (maxclique) separators, and claimed this modified algorithm retained the same time complexity. This algorithm has been used, for example, to recognize some classes of path graphs [2, 4].

2000 *AMS Subject Classification.* 68R10, 05C75, 05C85.

*Key Words and Phrases.* graph decomposition, clique separator, maximal clique, algorithm.

*Research supported in part by CAPES, CNPq, and FAPERJ.

†Research supported by FAPERJ (Bolsa Nota 10).

‡Research supported in part by CNPq and FAPERJ.

In this work, we provide a (minimal) counterexample to the correctness of the maxclique decomposition algorithm specified in [5], and propose a modification to it in order to obtain a correct algorithm which retains the $O(nm)$ complexity.

## 2 Definitions and preliminary results

Definitions and notations not specified here are standard and can be found in [1].

An *elimination ordering* of $G$ is a total ordering of $V(G)$. For ease of notation, we shall in general also treat an elimination ordering as a bijection between $V(G)$ and $\{1, \dots, n\}$. Given an elimination ordering $\pi$ of $G$, we say $u, v \in V(G)$ are *fillable* w.r.t. $\pi$ if they are nonadjacent and there exists a path $P = u, x_1, \dots, x_k, v$ in $G$ such that

$$\pi(x_i) < \min\{\pi(u), \pi(v)\}$$

for all $i \in \{1, \dots, k\}$. The set $F_\pi$ of *fill-in edges created by* $\pi$ is the set of all fillable pairs of vertices of $G$. A *minimal elimination ordering* (m.e.o.) of $G$ is an elimination ordering $\pi$ such that there is no other elimination ordering $\pi'$ of $G$ such that $F_{\pi'} \subset F_\pi$.

For an example of these concepts, see Figure 1. Note that the ordering presented in Figure 1(b) is not minimal.

Finding an elimination ordering with *minimal* (w.r.t. inclusion) set of fill-in edges can be done in $O(nm)$ time by a variation of lexicographic breadth-first search which is due to Rose, Tarjan, and Lueker [3]. This contrasts with the fact that the problem of finding an elimination ordering with *minimum* (w.r.t. cardinality) set of fill-in edges is NP-hard (Yannakakis 1981, cf. [5]).

Given an elimination ordering $\pi$ and a vertex $v$ of $G$, we define

$$C_\pi(v) = \{u \in V : \pi(u) > \pi(v) \text{ and } uv \in E \cup F_\pi\}.$$

Note that, once $F_\pi$ is known, determining $C_\pi(v)$ for all $v$ can be done in $O(m + |F_\pi|) = O(nm)$ time since, for each $uv \in E \cup F_\pi$, exactly one of

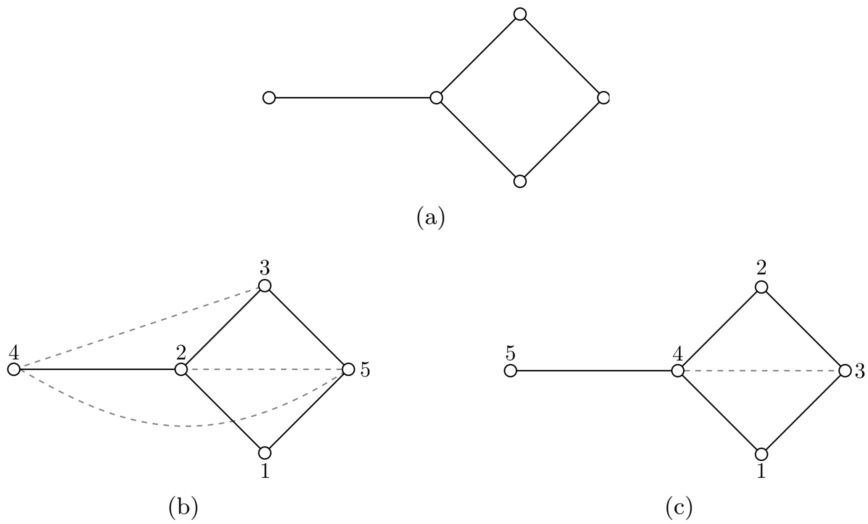


Figure 1: A graph (a) and two elimination orderings (b–c), along with the fill-in edges they create (represented by dashed lines).

$v \in C_\pi(u)$ or $u \in C_\pi(v)$ is true, so that a simple traversal of the set $E \cup F_\pi$ is enough to determine all of these sets.

We are now ready to specify Tarjan's clique decomposition algorithm (Algorithm 1 below), which uses those sets $C_\pi(v)$ that are cliques to separate a given graph.

---

**Algorithm 1:** Tarjan's clique decomposition algorithm

---

**Input**: A graph $G$.
**Output**: A decomposition of $G$ by clique separators, if one exists.
Compute an m.e.o. $\pi$ of $G$ ;
**foreach** $v \in V$ **do** compute $C_\pi(v)$ ;
**foreach** $v \in V$ *in increasing order w.r.t.* $\pi$ **do**
  **if** $C_\pi(v)$ *is a separating clique of* $G$ **then**
    $A(v) \leftarrow$ the vertex set of the conn. comp. of $G \setminus C_\pi(v)$ containing $v$;
    $$\left.\begin{array}{l} G_1 \leftarrow G[A(v) \cup C_\pi(v)]; \\ G_2 \leftarrow G \setminus A(v); \\ G \leftarrow G_2 \end{array}\right\} \text{“\textit{decomposition step}”}$$

---

Since each decomposition step can be performed in $O(m)$ time with, say, a breadth-first search, and since at most $n-1$ decomposition steps can separate $G$, the total running time of Algorithm 1 is $O(nm)$.

In order to obtain an algorithm that performed decomposition by maxclique separators, Tarjan proposed some modifications to Algorithm 1, resulting in Algorithm 2 below.

**Algorithm 2:** Tarjan's proposed maxclique decomposition algorithm

```
Input: A graph G.
Output: A decomposition of G by maxclique separators, if one exists.
Compute an m.e.o. π of G ;
foreach v ∈ V do compute C_π(v) ;
foreach v ∈ V in increasing order w.r.t. π do
    if C_π(v) is a separating clique of G then
        A(v) ← the vertex set of the conn. comp. of G∖C_π(v)
        containing v ;
        B(v) ← V∖(A(v) ∪ C_π(v)) ;
        if S = C_π(v) ∪ {v} is a maxclique of G and A(v) ≠ {v}, or
        ∃B' ⊂ B(v) s.t. S = C_π(v) ∪ B' is a maxclique of G then
            G_1 ← G[A(v) ∪ S];   }
            G_2 ← G[B(v) ∪ S];   }  "decomposition step"
            G ← G_2;             }
```

Unfortunately, Algorithm 2 is not correct. For a (minimal) counterexample, consider elimination ordering (c) of the graph of Figure 1. Note that since that graph is not chordal and only one fill-in edge is created, the ordering is minimal (indeed, it is *minimum*). We have $C_\pi(1) = C_\pi(2) = \{3,4\}$, $C_\pi(3) = \{4\}$, $C_\pi(4) = \{5\}$, and $C_\pi(5) = \emptyset$. Therefore, even though $G$ has two separating maxcliques, namely $\{1,4\}$ and $\{2,4\}$, none of these is found by the algorithm.

## 3 A new algorithm

**Algorithm 3:** Decomposition by Maxclique Separators, DMS

**Input**: A graph $G$.
**Output**: A decomposition of $G$ by maxclique separators, if one exists.
Compute an m.e.o. $\pi$ of $G$ ;
**foreach** $v \in V$ **do** compute $C_\pi(v)$ ;
**foreach** $v \in V$ *in increasing order w.r.t.* $\pi$ **do**
  **if** $C_\pi(v)$ *is a separating clique of* $G$ **then**
    $A(v) \leftarrow$ the vertex set of the conn. comp. of $G \setminus C_\pi(v)$ containing $v$ ;
    $B(v) \leftarrow V \setminus (A(v) \cup C_\pi(v))$ ;
    **if** $\exists A' \subset A(v)$ *s.t.* $C_\pi(v) \cup A'$ *is a maxclique of* $G$ *and* $A' \neq \emptyset$ **then**
      $G_1 \leftarrow G[A(v) \cup (C_\pi(v) \cup A')]$;
      $G_2 \leftarrow G[B(v) \cup (C_\pi(v) \cup A')]$;  *"type (i) decomposition step"*
      $G \leftarrow G_2$;
    **else if** $\exists B' \subset B(v)$ *s.t.* $C_\pi(v) \cup B'$ *is a maxclique of* $G$ **then**
      $G_1 \leftarrow G[A(v) \cup (C_\pi(v) \cup B')]$;
      $G_2 \leftarrow G[B(v) \cup (C_\pi(v) \cup B')]$;  *"type (ii) decomposition step"*
      $G \leftarrow G_2$;

**Theorem 3.1.** *Algorithm DMS has $O(nm)$ time complexity.*

**Sketch of proof.** For each $v \in V$, testing whether $C_\pi(v)$ is a separating clique of $G$ can be done in $O(m)$ time with, say, a breadth-first search. Computing $A'$ and $B'$, as defined in the algorithm, takes $O(m)$ time using a simple greedy procedure. Since this process is done at most $O(n)$ times, the total complexity is $O(nm)$. ■

The proof of correctness of Algorithm DMS is largely based on the following.

**Lemma 3.2** (Central Lemma)**.** *Let $C$ be a minimal separating clique of $G$, let $V_i$ be the vertex set of a connected component of $G \setminus C$, and let $v_i$ be the maximum vertex of $V_i$ w.r.t. $\pi$. If $v_i$ is not the maximum vertex of $G \setminus C$, then*

$$C_\pi(v_i) = C.$$

The proof of this result is similar to the one found in [5], and is omitted.

**Theorem 3.3** (Correctness, part 1)**.** *If $G$ has a maxclique separator, then for any minimal ordering $\pi$ of $G$, some decomposition step of an execution of Algorithm DMS separates $G$.*

**Sketch of proof.** Let $S$ be a separating maxclique of $G$, and let $S' \subseteq S$ be a minimal separating clique of $G$. By the Central Lemma, there exists at least one vertex $v$ of $G$ such that $C_\pi(v) = S'$. Using the notation of Algorithm DMS, if $S \smallsetminus S' \subset A(v)$, then in the worst case a decomposition step of type (i) at $v$ occurs. If $S \smallsetminus S' = A(v)$, then $G \smallsetminus S'$ has at least *three* connected components since $S$ separates $G$. Hence, again by the Central Lemma, there exists a vertex $u \in B(v)$ such that $C_\pi(u) = S'$, so that in the worst case a decomposition step of type (ii) at $u$ occurs. Otherwise, if $S \smallsetminus S' \subseteq B(v)$, then in the worst case a decomposition step of type (ii) at $v$ occurs. ∎

Note that Algorithm DMS works at each step by decomposing $G$ into two parts, and then discarding one of them. Therefore, it is necessary to prove that the discarded part contains no maxclique separators.

**Lemma 3.4.** *If $C_\pi(v)$ separates $G$, then $v$ is the maximum vertex of $A(v)$ w.r.t. $\pi$.*

**Sketch of proof.** Suppose, for a contradiction, that there exists $w \in A(v)$ s.t. $\pi(w) > \pi(v)$. If $v$ and $w$ are adjacent, then $w \in C_\pi(v)$, a contradiction since by definition $A(v) \cap C_\pi(v) = \emptyset$. Otherwise, by induction on the distance between $v$ and $w$, it can be shown that $v$, $w$ are fillable, which also implies $w \in C_\pi(v)$, a contradiction. ∎

**Theorem 3.5** (Correctness, part 2)**.** *For any graph $G$ and any minimal ordering $\pi$ of $G$, each decomposition step of Algorithm DMS creates at least one atom.*

**Sketch of proof.** We prove by induction on $k$ that if an execution of Algorithm DMS on a graph $G$ with ordering $\pi$ has $k$ decomposition steps, then each one creates an atom. If $k = 0$ then the result follows vacuously. Consider an execution that has $k$ decomposition steps, the first of which happens at vertex $v$ and separates $G$ into $G_1$ and $G_2$.

After this decomposition step, the execution proceeds in exactly the same way as a fresh execution on $G_2$ with the appropriate ordering, so that by the induction hypothesis each of the decomposition steps following the one at $v$ creates at least one atom.

Therefore, all that is left to prove is that $G_1$ is an atom. Let $M$ be the *first* separating maxclique of $G$ which was found by the algorithm (i.e., $M = C_\pi(v) \cup A'$ if the step was type (i), and $M = C_\pi(v) \cup B'$ if the step was type (ii)), and let $\pi_1$ be the restriction of $\pi$ to the vertices of $G_1$.

It is easy to see that $C_{\pi_1}(x) = C_\pi(x)$ for all $x \in A(v)$.

Now suppose for a contradiction that $G_1$ has a separating maxclique $S$ and let $S' \subseteq S$ be a minimal separating clique of $G_1$. At most one connected component of $G_1 \setminus S'$ can contain vertices of $M$, since it is a clique, and therefore, since by Lemma 3.4 all vertices of $A(v)$ come before all vertices of $C_\pi(v)$ in $\pi_1$, it follows by the Central Lemma that $S' = C_{\pi_1}(x)$ for some $x \in A(v)$. Furthermore, since $C_{\pi_1}(v)$ does not separate $G_1$, it follows that $x \neq v$, and therefore by Lemma 3.4 we have that $\pi_1(x) < \pi_1(v)$, a contradiction since in this case some decomposition step of Algorithm DMS on $G$ would have occurred at a vertex before $v$. ■

## Acknowledgements

The authors would like to thank one of the anonymous referees for the comments and suggestions which improved the presentation of this extended abstract.

## References

[1] J. A. Bondy and U. S. R. Murty. *Graph Theory.* Springer, 2007.

[2] C. L. Monma and V. K. Wei. Intersection graphs of paths in a tree. *Journal of Combinatorial Theory, Series B*, 41(2):141 – 181, 1986.

[3] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.

[4] J. P. Spinrad. *Efficient Graph Representations.* American Mathematical Society, Providence, RI, USA, 2003.

[5] R. E. Tarjan. Decomposition by clique separators. *Discrete Mathematics*, 55(2):221 – 232, 1985.

**Márcia R. Cerioli**
Instituto de Matemática e COPPE/Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro
Caixa Postal 68511, 21941-972, Rio de Janeiro, Brazil.

**Hugo Nobrega**
COPPE/Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro
Caixa Postal 68511, 21941-972, Rio de Janeiro, Brazil.

**Petrucio Viana**
Instituto de Matemática e Estatística
Universidade Federal Fluminense
Rua Mário Santos Braga s/n$^o$, 24020-140, Niterói, Brazil.

*Email addresses*: {cerioli, hugonobrega, petrucio}@cos.ufrj.br