Electronic Journal of Graph Theory and Applications **7** (1) (2019), 147–155

Electronic Journal of
Graph Theory and Applications

Maximum cycle packing using SPR-trees

Christin Otto, Peter Recht

Operations Research and Business Informatics
TU Dortmund
44221 Dortmund, Germany

christin-maria.otto@tu-dortmund.de, peter.recht@tu-dortmund.de

Abstract

Let $G = (V, E)$ be an undirected multigraph without loops. The maximum cycle packing problem is to find a collection $\mathcal{Z}^* = \{C_1, ..., C_s\}$ of edge-disjoint cycles $C_i \subset G$ of maximum cardinality $\nu(G)$. In general, this problem is $\mathcal{NP}$-hard. An approximation algorithm for computing $\nu(G)$ for 2-connected graphs is presented, which is based on splits of G. It essentially uses the representation of the 3-connected components of G by its SPR-tree. It is proved that for generalized series-parallel multigraphs the algorithm is optimal, i.e. it determines a maximum cycle packing $\mathcal{Z}^*$ in linear time.

Keywords: maximum cycle packing, decomposition, SPR-trees, edge-disjoint cycle
Mathematics Subject Classification : 05C38, 05C70
DOI: 10.5614/ejgta.2019.7.1.11

1. Introduction

Let $G = (V(G), E(G))$ be a finite and undirected graph with vertex set $V(G)$ and edge set $E(G)$ which may contain multiple edges but no loops. A graph $G' = (V', E')$ is a *subgraph* of G ($G' \subseteq G$), if $V' \subseteq V$ and $E' \subseteq E$. A subgraph $G' = (V', E') \subset G$ is *induced* by $E' \subset E$ ($G' = G|_{E'}$) if V' consists of all vertices that are incident with edges in E'. Similarly, $G' = (V', E') \subset G$ is *induced* by $V' \subset V$ ($G' = G|_{V'}$) if E' consists of all edges $e \in E$, that have both endvertices in V'. We will write $G \setminus V' := G|_{V \setminus V'}$ and $G \setminus E' := G|_{E \setminus E'}$, respectively. For $u \in V$ the *degree* $\delta_G(u)$ is the number of its incident edges in G. A path P of length $r \geq 0$ is a sequence of distinct edges $(e_1, \ldots, e_r)$ such that $e_i = (v_{i-1}, v_i) \in E(G)$ where the vertices $v_0, \ldots, v_r \in V(G)$

Received: 2 February 2017, Revised: 26 September 2018, Accepted: 23 December 2018.

are distinct. We sometimes say P is a v_0-v_r-path to emphasize the first and the last vertex of a path. A cycle C of length $r \geq 2$ is a sequence $(e_1, \ldots, e_{r-1}, e_r)$ such that $(e_1, \ldots, e_{r-1})$ is a path of length $r-1$ and $e_r = (v_{r-1}, v_0)$. Since P can be considered as a subgraph of G we sometimes say that P is induced by its edgeset $E(P)$. A graph G is connected if for each pair of vertices $v, w \in V$ there is a v-w-path in G. A set $S \subset V$ is called a *k-separator* of G ($k \geq 0$, $|S| = k$) if $G|_{V \setminus S}$ is not connected. A connected graph G is called *k-connected* if there is no $(k-1)$-separator in G. The maximum 1-connected subgraphs of G are called *1-components*. The maximum 2-connected subgraphs of G are called *blocks*. We say G is *k-separable* if there exist subgraphs G_1, G_2 of G such that $G = G_1 \cup G_2$ with $|V(G_1) \cap V(G_2)| = k$, $E(G_1) \cap E(G_2) = \emptyset$ and $|E(G_1)| \geq k, |E(G_2)| \geq k$. The pair $\{G_1, G_2\}$ is then called a *k-separation of G*. Two subgraphs $G' = (V', E')$ and $G'' = (V'', E'')$ are called *edge-disjoint* if $E' \cap E'' = \emptyset$. A *packing of edge-disjoint cycles* of cardinality s in G is a set $\mathcal{Z} = \{C_1, \ldots, C_s\}$ of cycles that are mutually edge-disjoint. A cycle packing $\mathcal{Z}^*$ of maximum cardinality is called a maximum cycle packing. Its cardinality $|\mathcal{Z}^*|$ is denoted by $\nu(G)$.

Packing edge-disjoint cycles in graphs is a classical graph-theoretical problem. There is a large amount of literature concerning cycle packing problems for example [12], [11], [10], [1], [20], [7], [6], [19], [18]. In [14], [2] and [8] simple approximation algorithms are described since cycle packing problems are typically hard [14].

The basic idea of this paper is to decompose G into suitable subgraphs G_i and relate maximum cycle packings $\mathcal{Z}_i$ of the G_i to a maximum cycle packing $\mathcal{Z}^*$ of G. In the case that G_i are the 1-components it holds that $\mathcal{Z}^* = \bigcup \mathcal{Z}_i$ and $\nu(G) = \sum \nu(G_i)$. If G is decomposed into blocks B_i it holds that $\nu(G) = \sum \nu(B_i)$. If G is 2-connected an appropriate tool to represent G by its 3-connected components is the SPR-tree [5]. In Section 2 this tool is used to obtain an algorithm that provides an approximation of a maximum cycle packing of G. The proof of optimality of the algorithm for general series-parallel graphs is given in Section 3.

2. Cycle packing by using SPR-trees

In [2] a greedy type algorithm was suggested for the determination of a large number of edge-disjoint cycles in an arbitrary graph G (see also [14]). Its basic idea is to search for the shortest cycle C in G, then delete it from G and delete also edges that cannot be contained in a cycle of $G \setminus C$. This procedure is continued until there are no edges left. The set of successively deleted cycles finally provides the approximation of a maximum cycle packing of G (Algorithm 1). The algorithm has approximation ratio $\mathcal{O}(\log n)$ (see [2]).

In the special case that G is 2-connected we, additionally, will exploit the *splits* of G into 3-components during the algorithmic procedure. By this we can relate the edge-disjoint cycles within each of these components to cycles in a cycle packing of G. Let G be a 2-connected multigraph and let $\{G_1, G_2\}$ be a 2-separation of G. If $\{u, v\} = V(G_1) \cap V(G_2)$, we call the 2-separation a *split*, if G_1 or G_2 has no 0- or 1-separator and $G_1 \setminus \{u, v\}$ or $G_2 \setminus \{u, v\}$ is non-empty and connected [16]. In [21] it was proved that 2-connected graphs that have no splits are either 3-connected or cycles of length ≥ 3 or a bundle of parallel edges between two vertices, respectively. For a split $\{G_1, G_2\}$ let G'_1 and G'_2 be the graphs obtained from G_1 and G_2 by adding an edge (u, v) to each of them where (u, v) is determined by the common vertices $\{u, v\} = V(G_1) \cap V(G_2)$. The added

Algorithm 1 Greedy algorithm for the maximum cycle packing problem

Require: Biconnected multigraph $G = (V, E)$ without loops.
Ensure: Cycle packing $\mathcal{C}$ of size $\underline{\nu}(G)$.

$\mathcal{C} \leftarrow \emptyset$ and $\underline{\nu}(G) \leftarrow 0$
while $G \neq \emptyset$ **do**
  for all vertices $v \in V$ with $\delta(v) \leq 1$ **do**
    delete v
  end for
  for all vertices $v \in V$ with $\delta(v) = 2$ **do**
    replace $e' = (u, v)$ and $e'' = (v, w)$ by $e = (u, w)$
  end for
  search for a shortest cycle $C \in G$
  $\mathcal{C} \leftarrow \mathcal{C} \cup C$
  $\underline{\nu}(G) \leftarrow \underline{\nu}(G) + 1$
  for all edges $e \in C$ **do**
    delete $e \in G$
  end for
end while
return Cycle packing $\mathcal{C}$ and lower bound $\underline{\nu}(G)$ of $\nu(G)$.

edges are called *virtual edges*. Since G_1' and G_2' are 2-connected one may repeat the split process as long as the obtained graphs admit splits. Each of the resulting graphs finally constructed in this way is called a *split component* of G. A split component contains edges from E and some virtual edges determined by its consecutive split operations. In [15] and [21] it was shown, that split components of G are uniquely determined and independent of the sequence in which consecutive split operations were performed.

By this G can be represented using the SPR-tree $\mathcal{T}(G) = (M, A)$ of G as defined in [3], which is an alternative to the definition of [5, 9]. If no ambiguity is possible we write $\mathcal{T}$ for short. A SPR-tree $\mathcal{T}$ of a 2-connected multigraph G is the smallest tree with the following properties

1. To every node[1] $\mu \in M$ a multigraph $G_\mu = (V_\mu, E_\mu)$ (called *skeleton* of μ) is associated.
2. Depending on their skeletons the nodes of $\mathcal{T}$ are of one of the following three types
   - μ is a S-node if G_μ is a cylce of length ≥ 3,
   - μ is a P-node if G_μ is a bundle of parallel edges,
   - μ is a R-node if G_μ is a simple 3-connected graph.
3. There is an edge $(\mu, \mu') \in A$ if and only if there is $u, v \in V$ such that G_μ and $G_{\mu'}$ have $\bar{e}_{(\mu,\mu')} := (u, v)$ as a common virtual edge.
4. The graph G can be recovered by applying the following operation on the nodes of $\mathcal{T}$: for $(\mu, \mu') \in A$ set $G_{(\mu,\mu')} := (G_\mu \setminus \bar{e}_{(\mu,\mu')}) \cup (G_{\mu'} \setminus \bar{e}_{(\mu,\mu')})$ and merge the two nodes μ, μ' to a new single node.

In [3] it was proved that a SPR-tree $\mathcal{T}$ of a 2-connected multigraph G exists and is unique. Moreover, it has neither two adjacent S-nodes nor two adjacent P-nodes. Since there is a strong

[1]The vertices in $\mathcal{T}$ are usually called nodes.

relation between SPR-trees and SPQR-trees introduced in [4], its size as well as the complexity of its determination is linear (in $\mathcal{O}(|V|+|E|)$) (cf. [3]).

In the sequel we assume that G is a 2-connected multigraph with no loops. Let $\mathcal{T}$ be the SPR-tree of G and μ be a leaf in $\mathcal{T}$ (i.e. a node in $\mathcal{T}$ such that $\delta_{\mathcal{T}}(\mu)=1$). The following approximation procedure applies Algorithm 1 in some of the iterations. It essentially exploits the SPR-tree representation of G and uses property 4 of $\mathcal{T}$ for an iterative construction of a large cycle packing $\mathcal{Z}$ in G. These cycles will be constructed from paths $\mathcal{P}_\mu$ for $\mu \in \mathcal{T}$. We initialize the sets $\mathcal{P}_\mu$ by $\mathcal{P}_\mu = \{P(e) \mid e \text{ is a real edge in } E_\mu\}$ with $P(e) := e$ and $\mathcal{Z}=\emptyset$.

During the procedure leaf nodes μ and the corresponding set $\mathcal{P}_\mu$ are successively *inspected*. Leaf nodes of S-type are always processed first, followed by R-leaves and P-leaves. Note, that for a leaf node $\mu \in M$ there is a unique node $\mu' \in M$ such that $(\mu,\mu') \in A$ and the edge set E_μ contains exactly one virtual edge $\bar{e}_{(\mu,\mu')}=(u,v)$. Within the procedure we set $pred(\mu) := \mu'$ the predecessor of μ. An inspection looks for the existence of edge-disjoint cycles on the real edges in E_μ. Such cycles correspond to edge-disjoint cycles in G. If there still remains an u-v-path on the real edges in E_μ there remains a corresponding u-v-path P_{uv} in G. In this case the virtual edge $\bar{e}_{(\mu,\mu')}$ in $E_{\mu'}$ is replaced by the real edge (u,v) and $P((u,v))$ is set to P_{uv}. If the virtual edge can not be replaced in such a way, it is deleted from $E_{\mu'}$.

Depending on the type of leaf node μ and its edge set E_μ the edge set $E_{\mu'}$ of $pred(\mu)$ is treated differently according to the following rules:

1_S μ is S-node: If the real edges in E_μ induce an u-v-path in E_μ, replace $\bar{e}_{(\mu,\mu')} \in E_{\mu'}$ by the real edge (u,v). Assign the u-v-path induced by $\bigcup\{E(P) \mid P \in \mathcal{P}_\mu\}$ to $P((u,v))$. Set $\mathcal{P}_{\mu'}$=$\mathcal{P}_{\mu'} \cup P((u,v))$, $\nu_\mu = 0$ and delete μ from $\mathcal{T}$.

2_R μ is R-node: Determine cycle packings $\mathcal{C}_1$ and $\mathcal{C}_2$ for the graphs induced by E_μ and $E_\mu \setminus \bar{e}_{(\mu,\mu')}$, respectively. Set $\nu_\mu = |\mathcal{C}_2|$, add the corresponding edge-disjoint cycles in G to $\mathcal{Z}$ and delete the related paths from $\mathcal{P}_\mu$. If $|\mathcal{C}_1| = |\mathcal{C}_2|$ delete $\bar{e}_{(\mu,\mu')} \in E_{\mu'}$. If $|\mathcal{C}_1| > |\mathcal{C}_2|$, there is an u-v-path in E_μ, not contained in any of the cycles of $\mathcal{C}_2$. Replace $\bar{e}_{(\mu,\mu')} \in E_{\mu'}$ by the real edge (u,v). Assign the u-v-path P_{uv} induced by $\bigcup\{E(P) \mid P \in \mathcal{P}_\mu\}$ to $P((u,v))$. Set $\mathcal{P}_{\mu'}$=$\mathcal{P}_{\mu'} \cup P((u,v))$ and delete μ from $\mathcal{T}$.

3_P μ is P-node:

(i) If $|E_\mu|$ is even, there is a cycle packing $\mathcal{C}_P$ with $\nu_\mu = \frac{|E_\mu|}{2} - 1$ cycles of length 2. Add the corresponding edge-disjoint cycles in G to $\mathcal{C}$. Then delete the related paths from $\mathcal{P}_\mu$. There remains an real edge e in E_μ, not contained in any of the cycles of $\mathcal{C}_P$. Replace $\bar{e}_{(\mu,\mu')} \in E_{\mu'}$ by the real edge (u,v) and assign the u-v-path P_{uv} induced by e to $P((u,v))$. Set $\mathcal{P}_{\mu'}$=$\mathcal{P}_{\mu'} \cup P((u,v))$ and delete μ from $\mathcal{T}$.

(ii) If $|E_\mu|$ is odd, there is a cycle packing $\mathcal{C}_P$ with $\nu_\mu = \frac{|E_\mu|-1}{2}$ cycles of length 2. Add the induced edge-disjoint cycles in G to $\mathcal{C}$. Further delete $\bar{e}_{(\mu,\mu')} \in E_{\mu'}$ and delete μ from $\mathcal{T}$.

The procedure terminates inspecting the final node:

Algorithm 2 Approximation algorithm for the maximum cycle packing problem

Require: Biconnected multigraph G without loops.
Ensure: Lower bound $\underline{\nu}(G)$ for the maximum cycle packing number $\nu(G)$.

$\mathcal{T}_G \leftarrow \mathrm{SPR}(G)$
$\mathcal{C} \leftarrow \emptyset$, $\underline{\nu}(G) \leftarrow 0$ and $\mathcal{P}_\mu \leftarrow \emptyset \quad \forall \mu \in M$
while $\exists$ SPR-node μ in $\mathcal{T}$ **do**
  for all S-leaves μ **do**
    $\mu' := pred(\mu)$
    if $\delta(v) = 2 \; \forall v \in V_\mu$ **then**
      replace $\bar{e}_{(\mu,\mu')} \in E_{\mu'}$ by real edge
      $\mathcal{P}_{\mu'} \leftarrow \mathcal{P}_{\mu'} \cup P(\bar{e}_{(\mu,\mu')})$
    end if
    $\nu_\mu \leftarrow 0$ and $\mathcal{T} \leftarrow \mathcal{T} \setminus \mu$
    $\underline{\nu}(G) \leftarrow \underline{\nu}(G) + \nu_\mu$
  end for
  for all R-leaves μ **do**
    $\mu' := pred(\mu)$
    $\mathcal{C}_1 \leftarrow$ Algorithm $1(G_\mu)$
    $\mathcal{C}_2 \leftarrow$ Algorithm $1(G_\mu \setminus \bar{e}_{(\mu,\mu')})$
    if $|\mathcal{C}_1| == |\mathcal{C}_2|$ **then**
      delete $\bar{e}_{(\mu,\mu')} \in E_{\mu'}$
    else if $|\mathcal{C}_1| > |\mathcal{C}_2|$ **then**
      replace $\bar{e}_{(\mu,\mu')} \in E_{\mu'}$ by real edge
      $\mathcal{P}_{\mu'} \leftarrow \mathcal{P}_{\mu'} \cup P(\bar{e}_{(\mu,\mu')})$
    end if
    $\nu_\mu \leftarrow |\mathcal{C}_2|$ and $\mathcal{T} \leftarrow \mathcal{T} \setminus \mu$
    $\underline{\nu}(G) \leftarrow \underline{\nu}(G) + \nu_\mu$ and $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_2$
  end for
  for all P-leaves μ **do**
    $\mu' := pred(\mu)$
    if $|E_\mu|$ is not even **then**
      $\nu_\mu \leftarrow \frac{|E_\mu|-1}{2}$
      delete $\bar{e}_{(\mu,\mu')}$ in $E_{\mu'}$
    else if $|E_\mu|$ is even **then**
      $\nu_\mu \leftarrow \frac{|E_\mu|}{2} - 1$
      replace $\bar{e}_{(\mu,\mu')}$ in $E_{\mu'}$ by real edge
      $\mathcal{P}_{\mu'} \leftarrow \mathcal{P}_{\mu'} \cup P(\bar{e}_{(\mu,\mu')})$
    end if
    $\underline{\nu}(G) \leftarrow \underline{\nu}(G) + \nu_\mu$ and $\mathcal{T} \leftarrow \mathcal{T} \setminus \mu$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\{P^{(2i-1)}, P^{(2i)}\} \mid P^{(i)} \in \mathcal{P}_\mu \forall i = 1, \ldots, \nu_\mu\}$
  end for
  for the final node μ **do**
    if μ is S-leaf and $\delta(v) = 2 \; \forall v \in V_\mu$ **then**
      $\nu_\mu \leftarrow 1$ and $\mathcal{C} \leftarrow \mathcal{C} \cup P|_{E(\bigcup_{e \in E_\mu} P(e))}$
    else if μ is R-leaf **then**
      $\mathcal{C}_1 \leftarrow$ Algorithm $1(G_\mu), \nu_\mu \leftarrow |\mathcal{C}_\infty|$ and $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_1$
    else if μ is P-leaf **then**
      $\nu_\mu \leftarrow \lfloor \frac{|E_\mu|}{2} \rfloor$ and $\mathcal{C} \leftarrow \mathcal{C} \cup \{\{P^{(2i-1)}, P^{(2i)}\} \mid P^{(i)} \in \mathcal{P}_\mu \forall i = 1, \ldots, \nu_\mu\}$
    end if
    $\underline{\nu}(G) \leftarrow \underline{\nu}(G) + \nu_\mu$ and $\mathcal{T} \leftarrow \mathcal{T} \setminus \mu$
  end for
end while

F If μ is the final node, determine a cycle packing $\mathcal{C}_F$ in the graph induced by E_μ. Set $\nu_\mu = |\mathcal{C}_F|$ and add the induced edge-disjoint cycles in G to $\mathcal{C}$.

Theorem 2.1. *Algorithm 2 determines a cycle packing $\mathcal{Z}$ of G of cardinality*

$$|\mathcal{Z}| = \sum_{\mu \in \mathcal{T}} \nu_\mu.$$

Proof. Let $\mathcal{T}$ be the SPR-tree of G.

When inspecting a S-node μ, the real edges in E_μ never induce a cycle, hence, $\nu_\mu = 0$. If the real edges induce an u-v-path in E_μ the corresponding u-v-path P_{uv} in G may contribute to an additional cycle C in $\mathcal{Z}$. Therefore, the virtual edge $\bar{e}_{(\mu,\mu')}$ in $E_{\mu'}$ is replaced by the real edge (u, v), $P((u, v))$ is set to P_{uv} and μ is deleted from $\mathcal{T}$. The possible cycle C might be determined when inspecting μ'.

When inspecting a R-node μ, two cycle packings $\mathcal{C}_1$ and $\mathcal{C}_2$ are determined for E_μ and $E_\mu \setminus \bar{e}_{(\mu,\mu')}$, respectively. E_μ induces at least a cycle packing of cardinality $\nu_\mu = |\mathcal{C}_2|$ in G. If $|\mathcal{C}_1| > |\mathcal{C}_2|$, $P((u, v))$ may also contribute to one more cycle C in $\mathcal{Z}$. Therefore the virtual edge $\bar{e}_{(\mu,\mu')}$ is replaced in $E_{\mu'}$ by (u, v) and a C might be determined when inspecting μ'.

When inspecting a P-node μ, different pairs of real edges in E_μ always induce edge-disjoint cycles in G. If $|E_\mu|$ is even, there are $\nu_\mu = \frac{|E_\mu|}{2} - 1$ of such pairs. The path P_{uv} induced by the remaining real edge may contribute to an additional cycle C in $\mathcal{Z}$. For this reason the virtual edge $\bar{e}_{(\mu,\mu')}$ is replaced in $E_{\mu'}$ by (u, v) and C might be determined when inspecting μ'. If $|E_\mu|$ is odd, there are $\nu_\mu = \frac{|E_\mu|-1}{2}$ pairs of real edges inducing the same number of additional cycles in $\mathcal{Z}$. □

Algorithm 2 has approximation ratio $\mathcal{O}(\log n)$, the same as Algorithm 1. If the SPR-tree $\mathcal{T}$ of G has no R-nodes we next proof in Section 3 that Algorithm 2 is optimal.

3. Proof of optimality for General Series-Parallel Graphs

Let G be a multigraph without loops. G is called generalized series-parallel, if it can be reduced to the K_2 by performing a sequence of simple operations:

(a) Replace two parallel edges by a single edge;

(b) replace two edges with a common incident node of degree 2 by a single edge;

(c) delete vertices of degree 1.

If there is no vertex of degree 1 to delete, G is called *series-parallel*. It is known that outerplanar graphs are generalized series-parallel [13]. A 2-connected generalized series-parallel multigraph G is reducable to K_2 by only performing operations (a) and (b). We will assume the input graph is 2-connected, since the algorithm could be launched on each block of G. The SPR-tree $\mathcal{T}$ of G has no R-nodes (cf. [17]). In this case the iterations of Algorithm 2 reflect a systematic sequence of operations of type (a) and (b) for the reduction of G. It leads to optimality of $\mathcal{Z}$.

Theorem 3.1. *Let G be a 2-connected, generalized series-parallel multigraph without loops. Then*

$$\nu(G) = \sum_{\mu \in \mathcal{T}} \nu_\mu,$$

i.e. Algorithm 2 determines a maximum cycle packing of G.

Proof. For the proof we will use induction on the number N of nodes in the SPR-tree $\mathcal{T}(G)$ of G.

Let $N = 1$, i.e. $\mathcal{T}(G)$ is either a P-node or a S-node, respectively. Hence, the series-parallel multigraph G is either a set of r parallel edges ($r \geq 3$) or a cycle of length ≥ 3. In the first case $\nu(G) = \lfloor \frac{r}{2} \rfloor$, in the second case $\nu(G) = 1$. In both cases $\nu(G)$ is the output of Algorithm 2 (step F).

Let $N \geq 2$ and let us assume that Algorithm 2 determines $\nu(G')$ for all series-parallel multigraphs G' such that $\mathcal{T}(G')$ has at most $N-1$ nodes. Let G be a series-parallel multigraph such that $\mathcal{T}(G)$ has N nodes. Now, we apply Algorithm 2. When selecting the *first* node $\mu \in \mathcal{T}(G)$ for inspection, the following cases can occur.

(a) μ is a S-leaf. Then Algorithm 2 treats μ according (1_S). The multigraph $G' = G \setminus (E_\mu \setminus \bar{e}_{(\mu,\mu')}) \cup (u,v)$ is series-parallel and $\mathcal{T}(G') = \mathcal{T}(G) \setminus \mu$, i.e. $\mathcal{T}(G')$ has $N-1$ nodes. Moreover, $\nu(G') = \nu(G)$. By hypothesis $\nu(G') = \sum_{\tilde{\mu} \in \mathcal{T}(G')} \nu_{\tilde{\mu}}$ and therefore $\sum_{\tilde{\mu} \in \mathcal{T}(G)} \nu_{\tilde{\mu}} = \sum_{\tilde{\mu} \in \mathcal{T}(G')} \nu_{\tilde{\mu}} + \nu_\mu = \nu(G') + 0 = \nu(G)$.

(b) μ is a P-leaf in $\mathcal{T}(G)$, then all leaf nodes are P-nodes.

(b1) There exists at least one leaf μ with an odd number of real edges, i.e. $|E_\mu|$ is even. Its predecessor μ' is a S-node. Algorithm 2 treats μ according $(3_P, (i))$. The multigraph $G' = G \setminus (E_\mu \setminus \bar{e}_{(\mu,\mu')}) \cup (u,v)$ is series-parallel and $\mathcal{T}(G') = \mathcal{T}(G) \setminus \mu$, i.e. $\mathcal{T}(G')$ has $N-1$ nodes. Moreover $\nu(G') = \nu(G) - (\frac{|E_\mu|}{2} - 1)$. By hypothesis $\nu(G') = \sum_{\tilde{\mu} \in \mathcal{T}(G')} \nu_{\tilde{\mu}}$ and, therefore, $\sum_{\tilde{\mu} \in \mathcal{T}(G)} \nu_{\tilde{\mu}} = \sum_{\tilde{\mu} \in \mathcal{T}(G')} \nu_{\tilde{\mu}} + \nu_\mu = \nu(G') + (\frac{|E_\mu|}{2} - 1) = \nu(G)$.

(b2) All P-leaves have an even number of real edges. Then a leaf μ is treated according $(3_P, (ii))$. Let $\mu' = pred(\mu)$. We assume that μ' is adjacent to $k \geq 1$ P-leaves $\mu_1, \ldots, \mu_k$ (let $\mu_1 = \mu$). Let $\hat{E}$ be the set of real edges in $\bigcup_{i \in \{1,\ldots,k\}} E_{\mu_i} \cup E_{\mu'}$. Then for the subgraph $\hat{G}$ induced by $\hat{E}$ we get $\nu(\hat{G}) = \sum_{i \in \{1,\ldots,k\}} \nu_{\mu_i}$. If $E \setminus \hat{E} = \emptyset$, $\mathcal{T}(G) = \mathcal{T}(\hat{G})$ and $\nu(G) = \sum_{i \in \{1,\ldots,k\}} \nu_{\mu_i} = \sum_{\tilde{\mu} \in \mathcal{T}(G)} \nu_{\tilde{\mu}}$. If $E \setminus \hat{E} \neq \emptyset$ then for the graph G' induced by $E \setminus \hat{E}$ we have $\nu(G') = \nu(G) - \sum_{i \in \{1,\ldots,k\}} \nu_{\mu_i}$. Now we show that G' is series-parallel. In $\mathcal{T}(G) \setminus (\bigcup_{i \in \{1,\ldots,k\}} \mu_i \cup \mu')$ μ' must have a predecessor $\mu'' = pred(\mu)$. μ'' is a P-node and must contain at least two parallel edges with endvertices, say u'', v''. One of them corresponds to the subgraph $\hat{G}$ when recovering G from $\mathcal{T}(G)$ (according to property 4). Since G is series-parallel, $G'' = G' \cup (u'', v'')$ is series-parallel. Since there is at least one more virtual edge parallel to (u'', v'') in $E_{\mu''}$, there must be a subgraph $\tilde{G} \subset G$ such that $\tilde{G}$ is reducible to a parallel edge of (u'', v'') and $E(\tilde{G}) \cap E(\hat{G}) = \emptyset$. According to property (b) of definition $G'' \setminus (u'', v'') = G'$ must be series-parallel. Obviously $\mathcal{T}(G') = \mathcal{T}(G) \setminus (\bigcup_{i \in \{1,\ldots,k\}} \mu_i \cup \mu')$. $\mathcal{T}(G')$ has $N-(k+1)$ nodes. By hypothesis $\nu(G') = \sum_{\tilde{\mu} \in \mathcal{T}(G')} \nu_{\tilde{\mu}}$ and $\nu(G) = \sum_{\tilde{\mu} \in \mathcal{T}(G')} \nu_{\tilde{\mu}} + \sum_{i \in \{1,\ldots,k\}} \nu_{\mu_i} = \sum_{\tilde{\mu} \in \mathcal{T}(G)} \nu_{\tilde{\mu}}$. □

The SPQR-tree of a 2-connected multigraph can be determined in linear time [9]. This holds also for the SPR-tree (see [3]) and we immediately get:

Corollary 3.1. *If G is a 2-connected, generalized series-parallel multigraph without loops, then a maximum cycle packing of G can be determined in linear time.*

References

[1] N. Alon, C. McDiarmid, and M. Molloy, Edge-disjoint cycles in regular directed graphs, *J. Graph Theory* **22** (1996), 231–237.

[2] A. Caprara, A. Panconesi, and R. Rizzi, Packing cycles in undirected graphs, *J. Algorithms* **48** (2003), 239–256.

[3] M. Chimani, Computing Crossing Numbers, PhD thesis, TU Dortmund (2008). `http://hdl.handle.net/2003/25955`

[4] G. Di Battista and R. Tamassia, Incremental planarity testing, *30th Annual Symposium on Foundations of Computer Science* (1989), 436–441.

[5] G. Di Battista and R. Tamassia, On-line maintenance of triconnected components with SPQR-trees, *Algorithmica* **15** (1996), 302–318.

[6] G. Dirac and P. Erdős, On the maximal number of independent circuits in a graph, *Acta Mathematica Academiae Scientiarum Hungarica* **14** (1963), 79–94.

[7] P. Erdős and L. Pósa, On the maximal number of disjoint circuits of a graph, *Publicationes Mathematicae Debrecen* **9** (1962), 3–12.

[8] Z. Friggstad and M.R. Salavatipour, Approximability of packing disjoint cycles, *Algorithmica* **60** (2011), 395–400.

[9] C. Gutwenger and P. Mutzel, A linear time implementation of SPQR-trees, *Proceedings of the 8th International Symposium on Graph Drawing* (2000), 77–90.

[10] L. Hao, Edge disjoint cycles in graphs, *J. Graph Theory* **13** (1989), 313–322.

[11] J. Harant, D. Rautenbach, P. Recht, and F. Regen, Packing edge-disjoint cycles in graphs and the cyclomatic number, *Discrete Math.* **310** (2010), 1456–1462.

[12] J. Harant, D. Rautenbach, P. Recht, I. Schiermeyer, and E.-M. Sprengel, Packing disjoint cycles over vertex cuts, *Discrete Math.* **310** (2010), 1974–1978.

[13] N.M. Korneyenko, Combinatorial algorithms on a class of graphs, *Discrete Appl. Math.* **54** (1994), 215–217.

[14] M. Krivelevich, Z. Nutov, M. R. Salavatipour, J. Verstraete, and R. Yuster, Approximation algorithms and hardness results for cycle packing problems, *ACM Transactions on Algorithms (TALG)* **3** (2007), 48.

[15] S. Mac Lane, A structural characterization of planar combinatorial graphs, *Duke Mathematical Journal* **3** (1937), 460–472.

[16] B. Mohar, Obstructions for the disk and the cylinder embedding extension problems, *Combinatorics, Probability and Computing* **3** (1994), 375–406.

[17] P. Mutzel, The SPQR-tree data structure in graph drawing, *Proceedings of the 30th International Conference on Automata, Languages and Programming* (2003), 34–46.

[18] D. Rautenbach and F. Regen, On packing shortest cycles in graphs, *Information Processing Letters* **109** (2009), 816–821.

[19] P. Recht and E.-M. Sprengel, Packing Euler graphs with traces, *Operations Research Proceedings 2011* (2012), 53–58.

[20] P. Recht and S. Stehling, On maximum cycle packings in polyhedral graphs, *Electron. J. Graph Theory Appl.* **2** (1) (2014), 18–31.

[21] W.T. Tutte, *Connectivity in graphs* (1966).