Computing sharp 2-factors in claw-free graphs

Hajo Broersma and Daniël Paulusma

Department of Computer Science, Durham University, DH1 3LE Durham,
United Kingdom, {hajo.broersma,daniel.paulusma}@durham.ac.uk

Abstract. In a recently submitted paper we obtained an upper bound for the minimum number of components of a 2-factor in a claw-free graph. This bound is sharp in the sense that there exist infinitely many claw-free graphs for which the bound is tight. In this paper we extend these results by presenting a polynomial algorithm that constructs a 2-factor of a claw-free graph with minimum degree at least four whose number of components meets this bound. As a byproduct we show that the problem of obtaining a minimum 2-factor (if it exists) is polynomially solvable for a subclass of claw-free graphs. As another byproduct we give a short constructive proof for a result of Ryjáček, Saito & Schelp.

1 Introduction

In this paper we consider 2-factors of claw-free graphs. Graph factors are well-studied. See [16] for a survey. Our motivation to study 2-factors goes back to the well-known NP-complete decision problem H-CYCLE (cf. [9]) in which the problem is to decide whether a given graph has a hamiltonian cycle, i.e., a connected 2-regular spanning subgraph. In the related problem 2-FACTOR the connectivity condition is dropped, hence the problem is to decide whether a given graph admits a 2-factor, i.e., a 2-regular spanning subgraph. This makes the problem considerably easier in the algorithmic sense: it is well-known that 2-FACTOR can be solved in polynomial time by matching techniques, and a 2-factor can be constructed in polynomial time if the answer is YES (cf [14]). Clearly, a hamiltonian cycle is a 2-factor consisting of one component, and the minimum number of components of a 2-factor can be seen as a measure for how far a graph is from being hamiltonian. So, from an algorithmic viewpoint a natural question is to consider the problem of determining a 2-factor of a given graph with a minimum number of components. Obviously, this is an NP-hard problem. Hence it makes sense to search for 2-factors with a reasonably small number of components if we aim for polynomial time algorithms. For this research we have restricted ourselves to the class of claw-free graphs. This is a rich class containing, e.g., the class of line graphs and the class of complements of triangle-free graphs. It is also a very well-studied graph class, both within structural graph theory and within algorithmic graph theory; see [7] for a survey. Furthermore, computing a 2-factor with a minimum number of components remains NP-hard for the class of claw-free graphs.

In a recently submitted paper [1] we already obtained an upper bound for the minimum number of components of a 2-factor in a claw-free graph. This bound is sharp in the sense that there exist infinitely many claw-free graphs for which the bound is tight; we will specify this later. When considering the related complexity problems, we soon realized that the proof methods used in [1] need to be extended in order to obtain a polynomial algorithm that constructs a corresponding 2-factor, e.g., a 2-factor whose number of components is at most our upper bound. In the present paper we present this polynomial time algorithm.

2 Terminology and background

We consider graphs that are finite, undirected and simple, i.e., without multiple edges and loops. For notation and terminology not defined in this paper we refer to [4].

Let $G = (V_G, E_G)$ be a graph of order $|G| = |V_G| = n$ and of size $e_G = |E_G|$. The neighbor set of a vertex x in G is denoted by $N_G(x) = \{y \in V_G \mid xy \in E_G\}$, and its cardinality by $d_G(x)$. We denote the minimum (vertex) degree of G by $\delta_G = \min\{d_G(x) \mid x \in V_G\}$. If no confusion can arise we often omit the subscripts.

Let K_n denote the complete graph on n vertices. A graph F is called a *2-factor* of a graph G if F is a 2-regular spanning subgraph of G, i.e., if F is a subgraph of G with $V_F = V_G$ and $d_F(x) = 2$ for all $x \in V_F$. A *claw-free* graph is a graph that does not contain an induced subgraph isomorphic to the four-vertex star $K_{1,3} = (\{u, a, b, c\}, \{ua, ub, uc\})$.

2.1 Known results

Several interesting problems are still open for claw-free graphs such as the conjecture of Matthews and Sumner [15] that every 4-connected claw-free graph is hamiltonian. However, there is quite a lot known on 2-factors in claw-free graphs, including some very recent results. Results of both Choudum & Paulraj [3] and Egawa & Ota [5] imply that every claw-free graph with $\delta \geq 4$ contains a 2-factor.

Theorem 1 ([3, 5]). *A claw-free graph with $\delta \geq 4$ has a 2-factor.*

We observe that every 4-connected claw-free graph has minimum degree at least four, and hence has a 2-factor. A 2-connected claw-free graph already has a 2-factor if $\delta = 3$ [20]. However, in general a claw-free graph with $\delta \leq 3$ does not have to contain a 2-factor. Examples are easily obtained.

Faudree et al. [6] showed that every claw-free graph with $\delta \geq 4$ has a 2-factor with at most $6n/(\delta + 2) - 1$ components. Gould & Jacobson [11] proved that, for every integer $k \geq 2$, every claw-free graph of order $n \geq 16k^3$ with $\delta \geq n/k$ has a 2-factor with at most k components. Fronček, Ryjáček & Skupień [8] showed that, for every integer $k \geq 4$, every claw-free graph G of order $n \geq 3k^2 - 3$ with $\delta \geq 3k - 4$ and $\sigma_k > n + k^2 - 4k + 7$ has a 2-factor with at most $k - 1$ components. Here σ_k denotes the minimum degree sum of any k mutually nonadjacent vertices.

If a graph G is claw-free, 2-connected and has $\delta \geq 4$, then G has a 2-factor with at most $(n+1)/4$ components [13]. If a graph G is claw-free, 3-connected and has $\delta \geq 4$, then G has a 2-factor with at most $2n/15$ components [13].

In [1] we considered claw-free graphs with $\delta \geq 4$. Our motivation for this is as follows. We first note that the number of components of a 2-factor in any graph on

n vertices is obviously at most $n/3$. For claw-free graphs with $\delta = 2$ that have a 2-factor we cannot do better than this trivial upper bound. This is clear from considering a disjoint set of triangles (cycles on three vertices). For claw-free graphs with $\delta = 3$ that have a 2-factor, the upper bound $n/3$ on its number of components is also tight, as shown in [1]. Hence, in order to get a nontrivial result it is natural to consider claw-free graphs with $\delta \geq 4$.

Our two main results in [1] provide answers to two open questions posed in [20].

Theorem 2 ([1]). *A claw-free graph G on n vertices with $\delta \geq 5$ has a 2-factor with at most $(n-3)/(\delta-1)$ components unless G is isomorphic to K_n.*

Theorem 3 ([1]). *A claw-free graph G on n vertices with $\delta = 4$ has a 2-factor with at most $(5n-14)/18$ components, unless G belongs to a finite class of exceptional graphs.*

Both results are tight in the following sense. Let $f_2(G)$ denote the minimum number of components in a 2-factor of G. Then in [20], for every integer $d \geq 4$, an infinite family $\{F_i^d\}$ of claw-free graphs with $\delta(F_i^d) \geq d$ is given such that $f_2(F_i^d) > |F_i^d|/d \geq |F_i^d|/\delta(F_i^d)$. This shows we cannot replace $\delta - 1$ by δ in Theorem 2. The bound in Theorem 3 is tight in the following sense. There exists an infinite family $\{H_i\}$ of claw-free graphs with $\delta(H_i) = 4$ such that

$$\lim_{|H_i| \to \infty} \frac{f_2(H_i)}{|H_i|} = \frac{5}{18}.$$

This family can be found in [20] as well.

The exceptional graphs of Theorem 3 have at most seventeen vertices. They are described in [1], and we will not specify them here. In [1] we also explain that Theorem 2 and 3 together improve the previously mentioned result of Faudree et al. [6] and that Theorem 2 also improves the previously mentioned result of Gould & Jacobson [11].

2.2 Results of this paper

The proofs in [1] do not yield algorithms for constructing 2-factors that satisfy the upper bounds in Theorems 2 and 3. In the remainder of this paper we will develop a new approach to these problems in order to establish polynomial algorithms that construct 2-factors of claw-free graphs with minimum degree at least four. Using our results in [1] we show that the number of components in these 2-factors are guaranteed to satisfy the upper bounds of Theorems 2 and 3. We will illustrate our approach by concentrating on Theorem 2, but the same approach works for Theorem 3 in exactly the same way. As a byproduct we show that the problem of obtaining a minimum 2-factor (if it exists) is polynomially solvable for a subclass of claw-free graphs which we describe later on. As another byproduct we give a short constructive proof for a result of Ryjáček, Saito & Schelp [19].

3 The algorithm for constructing 2-factors of claw-free graphs

We split the proof into six different parts. For the first two parts we do not have to develop any new theory or algorithms, but can rely on the beautiful existing machinery from the literature. The first part of this says that claw-free graphs behave the same with respect to our problem as line graphs obtained from them by performing some closure operation which will be explained shortly. The second part then describes the known equivalence of our problem with an analogous problem based on concepts and results in the preimage graphs of line graphs. Our new contributions are described and explained in the third, fourth, fifth and sixth part. In the third part we consider preimage graphs that are trees and in the fourth part we consider preimage graphs that are triangle-free. Finally, in the fifth and sixth part we translate the results back to the original domain of claw-free graphs and mention some special class for which our algorithm finds a 2-factor with a minimum number of components.

Step 1: restrict to line graphs of triangle-free graphs

The *line graph* of a graph H with edges $e_1, \ldots, e_p$ is the graph $L(H)$ with vertices $u_1, \ldots, u_p$ such that there is an edge between any two vertices u_i and u_j if and only if e_i and e_j share one end vertex in H. It is easy to verify and well-known (see e.g. [12]) that line graphs are claw-free graphs, but that the class of claw-free graph is much richer (in fact, line graphs have been characterized by a set of nine forbidden induced subgraphs). We show that we can restrict ourselves to an even smaller subclass of claw-free graphs, namely the class of line graphs of triangle-free graphs. For this purpose we use the *closure* concept as defined in [18].

The closure of a claw-free graph is defined as follows. Let G be a claw-free graph. Then, for each vertex x of G, the set of neighbors of x in G induces a subgraph with at most two components. If this subgraph has two components, both of them must be cliques. If the subgraph induced by $N(x)$ is connected, we add edges joining all pairs of nonadjacent vertices in $N(x)$. This operation is called the *local completion of G at x*. The *closure* $cl(G)$ of G is a graph we can obtain by recursively repeating the local completion operation, as long as this is possible. Ryjáček [18] showed that the closure of G is uniquely determined, i.e., that the ordering in which one performs the local completions does not matter. Ryjáček [18] also showed that G is hamiltonian if and only if $cl(G)$ is hamiltonian. This result was later extended to 2-factors [19].

Theorem 4 ([19]). *Let G be a claw-free graph. Then G has a 2-factor with at most k components if and only if $cl(G)$ has a 2-factor with at most k components.*

The following relationship between claw-free graphs and triangle-free graphs exists.

Theorem 5 ([18]). *If G is a claw-free graph, then there is a triangle-free graph H such that $L(H) = cl(G)$.*

It is well-known that apart from K_3 which is $L(K_3)$ and $L(K_{1,3})$, every connected line graph F has a unique H with $F = L(H)$ (see e.g. [12]). We call H the *preimage graph* of F. For K_3 we let $K_{1,3}$ be its preimage graph. For disconnected graphs we define the preimage graphs according to their components.

Recall that $f_2(G)$ denotes the minimum number of components in a 2-factor of a graph G. By Theorem 4 and Theorem 5, we deduce that for a claw-free graph G, $f_2(G) = f_2(cl(G)) = f_2(L(H))$, where H is the (triangle-free) preimage graph of $cl(G)$. Recall that the closure of a claw-free graph can be obtained in polynomial time. Since it is known that the preimage graph of a line graph can be obtained in polynomial (linear) time (see e.g. [17]) we can efficiently compute H.

Step 2: translate the problem into finding dominating systems

An *even* graph is a graph in which every vertex has a nonzero even degree. A connected even graph is called a *circuit*. For $q \geq 2$, a *star* $K_{1,q}$ is a complete bipartite graph with independent sets $A = \{c\}$ and B with $|B| = q$; the vertex c is called the *center* and the vertices in B are called the *leaves* of $K_{1,q}$.

Let H be a graph that contains a set $\mathcal{S}$ consisting of stars with at least three edges and circuits, all (stars and circuits) mutually edge-disjoint. We call $\mathcal{S}$ a *system that dominates* H or simply a *dominating system* if for every edge e of H the following holds:

- e is contained in one of the stars of $\mathcal{S}$, or
- e is contained in one of the circuits of $\mathcal{S}$, or
- e shares an end vertex with an edge of at least one of the circuits in $\mathcal{S}$.

Gould & Hynds [10] proved the following result.

Theorem 6 ([10]). *The line graph $L(H)$ of a graph H has a 2-factor with k components if and only if H has a dominating system with k elements.*

Combining Theorem 4 and Theorem 5 with Theorem 6 yields the following result.

Theorem 7. *Let G be a claw-free graph. Then G has a 2-factor with k components if and only if the (triangle-free) preimage graph of G has a dominating system with k elements.*

The *edge degree* of an edge xy in a graph H is defined as $d_H(x) + d_H(y) - 2$. We denote the minimum edge degree of H by $\delta_e = \delta_e(H)$. Due to the previous discussions it is clear that Theorem 2 is equivalent to the following theorem.

Theorem 8. *A triangle-free graph H with $\delta_e(H) \geq 5$ has a dominating system with at most $(e(H) - 3)/(\delta_e(H) - 1)$ elements unless H is isomorphic to $K_{1,e(H)}$.*

We will now concentrate on determining (in polynomial time) a *sharp dominating system*, i.e., one that satisfies the upper bound of Theorem 8. We first deal with the case that H is a tree. In this case we can even determine a minimum dominating system in polynomial time.

Step 3: Compute minimum dominating systems for trees

Here we present a polynomial time algorithm for computing the number of elements in a minimum dominating system of any given tree. We use the following new terminology. A *minimum dominating system*, or shortly, an *m-system* of a graph H is a dominating system of H with the smallest number of elements. We denote such a system by $\mathcal{M}(H)$, and its number of elements by $m(H)$. If H does not allow a dominating system we write $m(H) = \infty$.

A vertex with degree 1 in a graph F is called an *end vertex* or *leaf* of F. An edge which is incident with a leaf is called a *pendant* edge. We say that we *add a pendant edge* to F if we add a new vertex to F and join it to precisely one of the vertices of F. Two edges are called *independent* if they do not share any end vertices. A *matching* is a set of mutually independent edges.

We write $H^q(w)$ to denote a tree H that contains a vertex w to which we added q new pendant edges. Note that $H^0(w) = H$. Let $H_1, \ldots, H_p$ be a set of p mutually vertex-disjoint trees, where each H_i contains a vertex w_i. We say that we have *joined* trees $H_1, \ldots, H_p$ *in* $w_1, \ldots, w_p$ *by* u if we add a new vertex u with edges uw_i for $i = 1, \ldots, p$. If $p = 0$, then the resulting tree $H(u)$ is the single vertex u, which has a dominating system of 0 elements by definition. Before we present our algorithm we first deduce a number of equations. Note that $m(H^1(w)) = \infty$ if $H = (\{w\}, \emptyset)$.

Lemma 1. *Let $w_1, \ldots, w_p$ be a set of p vertices belonging to mutually disjoint trees $H_1, \ldots, H_p$, respectively. Let $H(u)$ be the tree obtained after joining $H_1, \ldots, H_p$ in $w_1, \ldots, w_p$ by u. Then $m(H(u)) =$*

$$\begin{cases} 0 & if\ p = 0 \\ \displaystyle\sum_{i=1}^{p} m(H^1(w_i)) & if\ p \in \{1,2\} \\ \min\Big\{ \displaystyle\sum_{i=1}^{p} m(H^1(w_i)), & \\ 1 + \displaystyle\min_{i_1<i_2<i_3} \big\{ \sum_{j=1}^{3} m(H_{i_j}) + \sum_{i \notin \{i_1,i_2,i_3\}} \min\{m(H_i), m(H^1(w_i))\}\big\} \Big\} & if\ p \geq 3. \end{cases}$$

Proof. We prove each case separately.

• Let $p = 0$. Then $H(u) = (\{u\}, \emptyset)$. So, $m(H(u)) = 0$ by definition of a dominating system.

• Let $1 \leq p \leq 2$. Then, in any dominating system of $H(u)$, u is not a star center, and consequently, each w_i is the center of a star containing the edge uw_i. Note that in each tree $H^1(w_i)$, w_i is a star center (because the new pendant edge to w_i needs to be covered by a star). Hence, we can combine any m-systems $\mathcal{M}^1(w_i)$ of each $H^1(w_i)$ to obtain an m-system $\mathcal{M}(H(u))$ with $\sum_{i=1}^{p} m(H^1(w_i))$ elements.

• Let $p \geq 3$. First consider the set of dominating systems of $H(u)$ in which u is not a star center. In all these dominating systems, each w_i is the center of a star containing the edge uw_i. Similar to the previous case, we can combine any m-systems of each

$H^1(w_i)$ to obtain a dominating system $\mathcal{S}$ of $H(u)$ with $\sum_{i=1}^{p} m(H^1(w_i))$ elements. We note that $\mathcal{S}$ has the minimum number of elements over all dominating systems of $H(u)$ in which u is not a star center.

Secondly, consider the set of dominating systems of $H(u)$ in which u is a star center. In all these dominating systems, the star with center u contains at least three edges, say uw_{i_1}, uw_{i_2}, and uw_{i_3}, by definition of a dominating system. For the remaining edges uw_i we act as follows. In each dominating system of $H(u)$ that has a star with center u, such an edge uw_i either belongs to the star with center u, or else to the star with center w_i. We compute an m-system $\mathcal{M}(H_i)$ and an m-system $\mathcal{M}(H^1(w_i))$. Then we choose the one with the smallest number of elements, which we denote by $\mathcal{M}_i^*$. So, $|\mathcal{M}_i^*| = \min\{m(H_i), m(H^1(w_1)\}$. We now combine the m-systems of H_{i_j} for $j = 1, 2, 3$, together with the dominating systems $\mathcal{M}_i^*$ and a star that contains the edges uw_{i_j} for $j = 1, 2, 3$ plus possibly some more edges depending on our choice for each $\mathcal{M}_i^*$. We try all possible triples (i_1, i_2, i_3), and choose the combination with the smallest total number of elements. This way we obtain a dominating system $\mathcal{S}'$ of $H(u)$ that has

$$1 + \min_{i_1<i_2<i_3} \big\{ \sum_{j=1}^{3} m(H_{i_j}) + \sum_{i \notin \{i_1,i_2,i_3\}} \min\{m(H_i), m(H^1(w_i))\} \big\}$$

elements. We note that $\mathcal{S}'$ has the minimum number of elements over all dominating systems of $H(u)$ in which u is a star center.

Finally, we compare the numbers of elements of $\mathcal{S}$ and $\mathcal{S}'$, and we choose (the) one with the smallest number of elements. This yields an m-system $\mathcal{M}(H(u))$. $\square$

Using Lemma 1 we can prove the following theorem.

Theorem 9. *The problem of finding a minimum dominating system is polynomially solvable for the class of trees.*

Proof. Let H be a tree with a designated vertex v^0. We partition $V(H)$ into $L_0 \cup L_1 \cup \ldots \cup L_r$ such that for $i = 0, \ldots, r$, L_i is the set of vertices at distance i from v^0. Note that $L_0 = \{v^0\}$. For $v \in V(H)\backslash\{v^0\}$, we let $v^+ \in N(v)$ be the first vertex on the (unique) path from v to v^0 in H, and we let the subtree H_v be the component of $H - vv^+$ that contains v.

Now let $v \in V(H)$. Suppose v has neighbors $w_1, \ldots, w_p$ in H_v. Then H_v is obtained after joining the p mutually disjoint trees $H_{w_1}, \ldots, H_{w_p}$ in $w_1, \ldots, w_p$ by v. Suppose we have already computed the values $m(H_{w_i})$ and $m(H^1_{w_i}(w_i))$. Then, using Lemma 1, we can easily compute $m(H_v)$. We observe that the tree $H^1_v(v)$ is obtained after joining the trees $H_{w_i}, \ldots, H_{w_p}$ together with a new single vertex tree $(\{w_{p+1}\}, \emptyset)$ in $w_1, \ldots, w_{p+1}$ by v. Hence, we can use Lemma 1 to compute $m(H^1_v(v))$ as well. So, our strategy is to recursively compute the values $m(H_v)$ and $m(H^1_v(v))$: for $i = 1, \ldots, r$, we first compute the values $m(H_{v^i})$ and $m(H^1_{v^i}(v^i))$ for all $v^i \in L_i$, and use them to compute $m(H_{v^{i-1}})$ and $m(H^1_{v^{i-1}}(v^{i-1}))$ for all $v^{i-1} \in L_{i-1}$ according to Lemma 1. Clearly, computing $m(H)$ this way can be done in polynomial time.

In order to find an m-system $\mathcal{M}(H)$, we keep track of the stars as follows. Firstly, for each $v \in V(H)$, we remember whether v is a star center in an m-system of H_v

when we compute $m(H_v)$. In case v is the center of a star S_v, we keep track of the edges in S_v as well. Secondly, we check whether v becomes the center of a star S_v (and which edges belong to S_v if S_v exists) both when we compute $m(H_{v^+})$ and when we compute $m(H^1_{v^+}(v^+))$. Note that we can do this in polynomial time when we use the formula in Lemma 1. With the above information we can efficiently compute an m-system $\mathcal{M}(H)$, as the following claim shows.

Claim. For all v in each L_i we can compute in polynomial time whether v is the center of a star S_v of an m-system $\mathcal{M}(H)$ and, if so, which edges of H belong to S_v.

We prove this claim by induction on i. Let $i = 0$. When we computed the value for $m(H_{v^0}) = m(H)$ by using Lemma 1, we checked whether v^0 is the center of a star in an m-system of H. In case v^0 is the center of such a star S_{v^0}, we also remembered which edges belong to S_{v^0}.

Now suppose $i \geq 1$. Let $v \in L_i$. By the induction hypothesis, we know whether v^+ is the center of a star in an m-system $\mathcal{M}(H)$ or not. First suppose v^+ is not the center of a star in an m-system $\mathcal{M}(H)$. Then v is the center of a star S_v in $\mathcal{M}(H)$, and S_v is a star in an m-system $\mathcal{M}(H^1_v(v))$ as well. So, we kept track of S_v. Now suppose v^+ is the center of a star S_{v^+} in an m-system $\mathcal{M}(H)$. By the induction hypothesis, we know which edges S_{v^+} has. Then there are two cases: either vv^+ belongs to S_{v^+}, or it does not. If vv^+ belongs to S_{v^+}, then v is the center of a star S_v in $\mathcal{M}(H)$ if and only if S_v is a star in $\mathcal{M}(H_v)$. If vv^+ does not belong to S_{v^+}, then v is the center of a star S_v in $\mathcal{M}(H)$, and S_v is a star in an m-system $\mathcal{M}(H^1_v(v))$. In both cases we kept track of all the edges of S_v. $\square$

Step 4: Compute sharp dominating systems for general triangle-free graphs

Suppose G is a claw-free graph. Let H be the preimage of $cl(G)$, i.e., the triangle-free graph with $L(H) = cl(G)$. We now assume that H is not a tree.

The key idea behind our approach in this case is to start with an even subgraph X of H, then to "break" the circuits in X by removing a number of edges, such that we obtain a new graph H^* that is a forest. Then we can apply our approach from the previous section to each component of H^* if we first add sufficiently many pendant edges to ensure that each component has minimum edge-degree at least $\delta_e(H)$. In this procedure we have to add more edges than we remove. However, we will have the following advantage. The added pendant edges have to be dominated by (extra) stars in any dominating system of H^*, and these stars can be merged together into fewer elements of a dominating system in the original graph H. In other words, the larger number of stars we get by applying the upper bounds to H^* will provide the necessary compensation for the larger number of edges that we created. This way we are able to establish our upper bound for H. In [1] we gave a nonconstructive proof to show that this approach works. This proof in [1] was based on a number of assumptions on the choice of the even subgraph X of H. Here we follow an alternative approach which enables a constructive proof.

Let X be an even subgraph of H with set of components $\mathcal{C}$. Let $\mathcal{C}_4 \subset \mathcal{C}$ be the set of components of order 4. For each C in $\mathcal{C}$ we choose an edge e_C of C and for each C in $\mathcal{C}_4$ we choose two independent edges e_C, e'_C of C. We call the set of all chosen edges the

X*-set* and denote it by M. Note that M is a matching of H. Let $H^* = (H - E(X)) \cup M$. We call H^* the X*-graph*.

Lemma 2. *We can compute an even subgraph X of H such that H^* is a forest in polynomial time.*

Proof. We use an algorithm based on the following arguments:
Phase 1. We first construct an even subgraph X' of H. We can do this in polynomial time by adding mutually edge-disjoint cycles to X' until this is not possible anymore.
Phase 2. We choose an X'-set M' and check (in polynomial time) whether its X'-graph H' is a forest. If it is a forest, then we are done.

Suppose H' is not a forest. Let D be a cycle in H'. Consider the graph $X' \cup D$. For each e in $E(X' \cap D)$ we do as follows. Let e belong to a circuit C of X. Then C shares at most two edges with D. If C only shares e with D then we remove e from $X' \cup D$. In the other case C belongs to $\mathcal{C}_4$ and we remove the two edges of C that are not on D. This way we obtain an even subgraph Y' of H in polynomial time. We go to Phase 2 with Y' instead of X'.

We show that either $e(Y') > e(X')$ or else, if $e(Y') = e(X')$, then Y' contains fewer components than X'. This means that the algorithm will terminate at a certain moment with our desired graph X.

The above can be verified as follows. Note that we removed exactly $e(X' \cap D)$ edges and we added $e(D \setminus X')$ edges. So we are done if $e(D \setminus X') > e(X' \cap D)$. Since M' is a matching, we find that $e(X' \cap D) \leq e(D)/2$, so $e(D) = e(X' \cap D) + e(D \setminus X') \leq e(D)/2 + e(D \setminus X')$, so $e(D \setminus X') \geq e(D)/2 \geq e(X' \cap D)$. We are done unless $e(D \setminus X') = e(X' \cap D) = e(D)/2$.

Suppose the latter is the case. Then we are done if Y' contains fewer components than X'. Suppose Y' and X' have the same number of components. Then $X' \cap D$ belongs to exactly one circuit of X'. We already deduced that $e(X' \cap D) = e(D)/2 \geq 2$. Hence D is a four-cycle, but then the triangle-free graph H contains an induced K_4. With this contradiction we have completed the proof of this claim. □

The remainder of the constructive proof is exactly the same as the corresponding parts in our nonconstructive proof in [1]. We do not include it here due to the page restriction.

Step 5: translate the dominating systems back into 2-factors

Once we have obtained a dominating system $\mathcal{S}$ for the preimage graph H with $cl(G) = L(H)$, it is easy to translate this back into a 2-factor of $cl(G)$ in polynomial time:

- the stars in $\mathcal{S}$ correspond to complete graphs in $cl(G)$ on at least three vertices; a hamiltonian cycle can clearly be constructed in polynomial time;
- the circuits in $\mathcal{S}$ and the edges they dominate correspond to hamiltonian subgraphs in $cl(G)$; one can construct a hamiltonian cycle by traversing the circuit, picking up the edges (vertices in $cl(G)$) one by one and inserting dominated edges at the first instance an end vertex of a dominated edge is encountered. For traversing the circuits we use the polynomial algorithm that finds a eulerian tour in an even connected graph (cf. [4]).

Step 6: translate 2-factors in $cl(G)$ to 2-factors in G

We first introduce some notations. Let $C = v_1v_2 \dots v_pv_1$ be a cycle with a fixed orientation. The successor v_{i+1} of v_i is denoted by $v_i^{+C} = v_i^+$ and its predecessor v_{i-1} by $v_i^{-C} = v_i^-$. The segment $v_iv_{i+1}\dots v_j$ is denoted by $v_i\overrightarrow{C}v_j$, where the subscripts are to be taken modulo $|C|$. The converse segment $v_jv_{j-1}\dots v_i$ is denoted by $v_j\overleftarrow{C}v_i$. We use similar notations for paths.

We assume we are given a 2-factor F' of $cl(G)$ of a claw-free graph G. Let k be the number of components of F'. Here, we show how to obtain in polynomial time a 2-factor F of G such that F has *at most* k components. We base our translation of the following new theorem, which generalizes a similar result for hamiltonicity [2] in algorithmic sense.

Theorem 10. *Let G be a graph and let $\{u, v, x, y\}$ be a subset of four vertices of V_G such that $uv \notin E_G$ and $\{x, y\} \subseteq N(u) \cap N(v)$. Let $N(x) \subseteq N(u) \cup N(v) \cup \{u, v\}$ and let $N(y) \setminus (N(x) \cup \{x\})$ induce a complete graph (or be empty). Then we can find a 2-factor of G with at most k components in polynomial time, if $G + uv$ has a 2-factor with k components.*

Proof. Suppose $G + uv$ has a 2-factor F' with at most k components. Below we give a number of polynomial time transformations of F' such that we obtain a 2-factor F of G with at most k components. If $uv \notin E_{F'}$ then F' is a 2-factor of G. Suppose $uv \in E_D$ for some (cycle) component D of F', say $v = u^-$. Let $P = u\overrightarrow{D}v$. We distinguish the following three cases.

First suppose $x \notin V_D$. Let $x \in V_{D'}$ for some (cycle) component D' of F. By our assumptions, we may without loss of generality assume that $x^{+D'}u \in E_G$. Then we replace D and D' by a new cycle $ux^{+D'}\overrightarrow{D'}xv\overleftarrow{P}u$, and we are done.

Second suppose $x \in V_D$ but $y \notin V_D$. Let $y \in V_{D^*}$ for some (cycle) component D^* of F. Let $y' = y^{+D^*}$ and $y'' = y^{-D^*}$ be the neighbors of y on D^*. Suppose $y'y'' \in E_G$. Then we replace D^* by $y'\overrightarrow{D^*}y''y'$ and D by $uyv\overleftarrow{P}u$, and we are done. Suppose $y'y'' \notin E_G$. Since $N(y)\setminus(N(x)\cup\{x\})$ induces a complete graph, we find that one of the edges xy', xy'', say xy', must exist in G. By our assumptions, we then find that $y'u$ or $y'v$ belongs to E_G, and we are done by the same argument as in the previous case.

Third suppose $\{x, y\} \subset V_D$. Say x is on $u\overrightarrow{P}y$. First suppose $xy \in E_D$. We replace D by $u\overrightarrow{P}xv\overleftarrow{P}yu$, and we are done. Now suppose $xy \notin E_D$. Then $x^+ \neq y$. By our assumptions, $x^+ \in N(u) \cup N(v)$. Suppose $ux^+ \in E_G$. We replace D by $ux^+\overrightarrow{P}vx\overleftarrow{P}u$. Hence we may assume $vx^+ \in E_G$. Suppose $y^- = x^+$. Then we replace D by $uy\overrightarrow{P}vy^-\overleftarrow{P}u$. Hence we may assume $y^- \neq x^+$. Suppose $y^-x \in E_G$. Then we replace D by $u\overrightarrow{P}xy^-\overleftarrow{P}x^+v\overleftarrow{P}yu$. Hence we may assume $y^-x \notin E_G$. Suppose $y^+ = v$. Then we replace D by $u\overrightarrow{P}xvx^+\overrightarrow{P}yu$. Hence we may assume $y^+ \neq v$. Suppose $y^+x \in E_G$. Then we replace D by $u\overrightarrow{P}xy^+\overrightarrow{P}vx^+\overrightarrow{P}yu$. Hence we may assume $y^+x \notin E_G$. As $y^-x \notin E_G$, we then find $y^-y^+ \in E_G$ due to our assumptions. Then we replace D by $u\overrightarrow{P}y^-y^+\overrightarrow{P}vyu$. This proves Theorem 10. $\square$

Note that in Theorem 10, x and y can be nonadjacent, and G does not have to be claw-free. However, the following observation is easy to see.

Observation 1 ([2]) *If G is claw-free, then the conditions of Theorem 10 are satisfied if x and y are adjacent.*

Then, by the following observation, we can indeed transform a 2-factor of $cl(G)$ that has k components to a 2-factor of G that has at most k components. This means we have proven our main result. For convenience we include the proof of the next observation.

Observation 2 ([2]) *Let x be a vertex of a claw-free graph G with $G[N(x)]$ connected and non-complete. Then the local completion of G at x can be obtained by iteratively joining pairs $\{u, v\} \subseteq N(x)$ that satisfy the conditions in Theorem 10 for some $y \in N(u) \cap N(v)$.*

Proof. Consider the subgraph H_x of G induced by $N(x) \cup \{a \in V_G \mid ab \in E_G$ for some $b \in N(x)\}$. Note that x is a vertex of H_x and that H_x is claw-free. Hence, by Observation 1, x and y satisfy the conditions of Theorem 10 (in H_x) for every $y \in N(x)$. Since we only join nonadjacent pairs in $N(x)$, $N(x)$ and $N(y)$ will keep these properties for all $y \in N(x)$. □

Note that the above approach gives a short constructive proof for Theorem 4 (the result of Ryjáček, Saito & Schelp in [19]).

We note that Theorem 9 has the following consequence as a byproduct. We need a few definitions before we can state the result. A *cut vertex* of a graph G is a vertex whose removal increases the number of components. A *block* of G is a maximal subgraph of G without cut vertices (of itself). Hence if G has no isolated vertices, its blocks are either K_2s or (maximal) 2-connected subgraphs. For the purpose of our next result we call a block B of a claw-free graph G a *semiclique* if B becomes a complete subgraph of $cl(G)$. Since a claw-free graph in which every block is a semiclique has a forest as its preimage, we obtain the following consequence of Theorem 9.

Corollary 1. *Let G be a claw-free graph in which all blocks are semicliques. If G has a 2-factor, then we can construct a minimum 2-factor of G in polynomial time.*

4 Conclusions

In a recently submitted paper we obtained sharp upper bounds for the minimum number of components of a 2-factor in a claw-free graph. Here we extended these results by presenting a polynomial algorithm that constructs a 2-factor of a claw-free graph with minimum degree at least four whose number of components meets this bound. As a byproduct we showed that the problem of obtaining a minimum 2-factor (if it exists) is polynomially solvable for a subclass of claw-free graphs in which all blocks are semicliques. As another byproduct we gave a short constructive proof for a result of Ryjáček, Saito & Schelp.

Our polynomial time algorithm yields a 2-factor with a number of components below a guaranteed upper bound. This upper bound is completely determined by an upper bound we find for the number of elements of a dominating system of a certain tree (that is obtained from he corresponding triangle-free graph in Theorem 8). As this upper bound is sharp (cf. [20]), our next goal will be to determine the extremal tree cases and try to exclude these from happening. This refined analysis may lead to a better upper bound for claw-free graphs for which the current upper bound is not sharp. Another way to improve our algorithm is trying to refine the algorithm that constructs the tree H^* in Lemma 2 such that we have more information on the number of circuits in the even subgraph X of H.

Finally, Corollary 1 shows that our algorithm yields a 2-factor with a minimum number of components for claw-free graphs with arbitrary minimum degree in which all blocks are semicliques. In future research we aim to generalize this result, i.e, to find a larger class of claw-free graphs for which our (possibly modified) algorithm solves the problem of finding a minimum 2-factor. We will also analyze the class of claw-free graphs with minimum degree 3 that have a 2-factor more carefully.

References

1. H.J. Broersma, D. Paulusma and K. Yoshimoto, *Sharp upper bounds for the minimum number of components of 2-factors in claw-free graphs*, submitted. see http://www.dur.ac.uk/daniel.paulusma/Papers/Submitted/claw.pdf
2. H.J. Broersma and H. Trommel, *Closure concepts for claw-free graphs*, Discrete Math. **185** (1998) 231–238.
3. S.A. Choudum and M.S. Paulraj, *Regular factors in $K_{1,3}$-free graphs*, J. Graph Theory **15** (1991) 259–265.
4. R. Diestel, *Graph Theory, Second edition*, Graduate Texts in Mathematics **173**, Springer (2000).
5. Y. Egawa and K. Ota, *Regular factors in $K_{1,n}$-free graphs*, J. Graph Theory **15** (1991) 337–344.
6. R.J. Faudree, O. Favaron, E. Flandrin, H. Li, Z. Liu, *On 2-factors in claw-free graphs*, Discrete Math. **206** (1999) 131–137.
7. R. Faudree, E. Flandrin, and Z. Ryjáček *Claw-free graphs—a survey*, Disc. Math. **164** (1997) 87–147.
8. D. Fronček, Z. Ryjáček, and Z. Skupień, *On traceability and 2-factors in claw-free graphs*, Discussiones Mathematicae Graph Theory **24** (2004) 55–71.
9. M.R. Garey and D.S. Johnson, *Computers and Intractability*. W.H. Freeman and Co., New York, 1979.
10. R. Gould and E. Hynds, *A note on cycles in 2-factors of line graphs*, Bull. of ICA. 26 (1999) 46–48.
11. R.J. Gould and M.S. Jacobson, *Two-factors with few cycles in claw-free graphs*, Discrete Math. **231** (2001) 191–197.
12. F. Harary, *Graph Theory*, Addison-Wesley, Reading MA, 1969.
13. B. Jackson and K. Yoshimoto, *Even subgraphs of bridgeless graphs and 2-factors of line graphs*, Discrete Math. **307** (2007) 2775–2785.
14. L. Lovasz and M.D. Plummer, *Matching Theory*, North-Holland Mathematics Studies 121, North-Holland, Amsterdam.
15. M.M. Matthews and D.P. Sumner, *Hamiltonian results in $K_{1,3}$-free graphs*, J. Graph Theory **8** (1984) 139–146.
16. M.D. Plummer, *Graph factors and factorization: 1985-2003: A survey*, Discrete Math. **307** (2007) 791–821.
17. N.D. Roussopoulos, *A* $\max\{m,n\}$ *algorithm for determining the graph H from its line graph G*, Information Processing Letters **2** (1973) 108-112.
18. Z. Ryjáček, *On a closure concept in claw-free graphs*, J Combin Theory Ser B **70** (1997) 217–224.
19. Z. Ryjáček, A. Saito and R.H. Schelp, *Closure, 2-factor, and cycle coverings in claw-free graphs*, J. Graph Theory **32** (1999) 109–117.
20. K. Yoshimoto, *On the number of components in a 2-factor of a claw-free graph*, Discrete Math. **307** (2007) 2808–2819.

Appendix A

Here we describe the procedure that was mentioned in Step 4 in more detail. Let X be an even subgraph of H such that H^* is a forest. Let $\mathcal{C}$ be the set of components of X and let $d = \delta_e(H)$. Since H is triangle-fee, $|C| \geq 4$ for all $C \in \mathcal{C}$. We consider each $C \in \mathcal{C}$ separately, and distinguish two cases. For each case we determine the net increase in the number of edges in order to restore the minimum edge-degree.

Case 1 $|C| = 4$.
Then C is a cycle on four vertices, since H is triangle-free. Let $C = u_1u_2u_3u_4u_1$. We remove the edges u_1u_4 and u_2u_3. For each u_i, we add $d - d(u_i) + 2$ new pendant edges. Clearly, $d(u_i) + d(u_{i+1}) - 2 \geq d$ for each $i \in \{1, 2, 3, 4\}$ (where $u_5 = u_1$). Then we find

$$\sum_{i=1}^{4} d(u_i) - 4 \geq 2d.$$

Hence the net increase in the number of edges is

$$\sum_{i=1}^{4} (d - d(u_i) + 2) - 2 \leq 4d - (2d + 4) + 8 - 2 = 2d + 2. \tag{1}$$

Case 2. $|C| \geq 5$.
Let $V(C) = \{u_1, \ldots, u_\ell\}$, where we assume that u_1u_2 is an edge. We remove the edges of $E(C) \setminus \{u_1u_2\}$. For $i = 1, 2$, we add $d - d(u_i) + 2$ new pendant edges. For $i \geq 3$, we add $d - d(u_i) + 3$ new pendant edges. Since $d(u_i) + d(u_{i+1}) - 2 \geq d$, we get

$$\sum_{i=1}^{\ell} d(u_i) - l \geq \frac{\ell d}{2}.$$

Hence the net increase in the number of edges is at most

$$\sum_{i=1}^{\ell} (d - d(u_i) + 2) + (\ell - 2) - (\ell - 1) \leq \ell d - (\frac{\ell d}{2} + \ell) + 2\ell - 1 = \frac{\ell d}{2} + \ell - 1 \tag{2}$$

After we have performed one of the above operations as considered in Case 1 or Case 2 for every circuit in X, we have obtained a graph H^*. By Claim 2, we can guarantee that H^* is a forest. Due to our construction, $\delta_e(H^*) \geq d$ and $e(H^*) \geq 2d+2$. Hence, we can apply the approach from the previous section to each component of H^* to obtain a minimum dominating system $\mathcal{S}^*$.

Now we can use the following two lemmas from [1].

Lemma 3. *A tree H with $\delta_e(H) \geq 5$ has a dominating system $\mathcal{S}$ such that the set of centers of stars in $\mathcal{S}$ is $\bigcup_{i \geq \frac{\delta_e(H)}{2}+1} V_i(H)$.*

Lemma 4. *If H is a tree with $\delta_e(H) \geq 2$, then*

$$\sum_{i \geq \frac{\delta_e(H)}{2}+1} |V_i(H)| \leq \max\{1, \frac{e(H) - 3}{\delta_e(H) - 1}\}.$$

Using these two lemmas we know:

$$|\mathcal{S}^*| \leq \frac{e(H^*)-3}{\delta_e(H^*)-1} \leq \frac{e(H^*)-3}{d-1}. \tag{3}$$

For every vertex u in any $C \in \mathcal{C}$, there exists a star $A_u \in \mathcal{S}^*$ whose center is u, since u is adjacent to a leaf of H^*. Therefore,

$$\mathcal{S} = (\mathcal{S}^* \setminus \bigcup_{C \in \mathcal{C}} \{A_u : u \in C\}) \cup \mathcal{C}$$

is a system that dominates H, because every edge in $E(A_u) \cap E(H)$ is incident with u and there exists a circuit in $\mathcal{C}$ passing through u.

In order to complete our analysis we will calculate the cardinality of $\mathcal{S}$. Let $\mathcal{C}_i$ be the set of circuits with i vertices of X. By (1) and (2), we obtain

$$e(H^*) \leq e(H) + (2d+2)|\mathcal{C}_4| + \sum_{i \geq 5}(\frac{id}{2} + (i-1))|\mathcal{C}_i|. \tag{4}$$

For each $C \in \mathcal{C}_i$, we removed i stars and added one circuit from $\mathcal{S}^*$. This, together with inequalities (3) and (4), implies

$$\begin{aligned}
|\mathcal{S}| &\leq |\mathcal{S}^*| - 3|\mathcal{C}_4| - \sum_{i\geq 5}(i-1)|\mathcal{C}_i| \\
&\leq \frac{e(H^*)-3}{d-1} - 3|\mathcal{C}_4| - \sum_{i \geq 5}(i-1)|\mathcal{C}_i| \\
&\leq \frac{e(H) + (2d+2)|\mathcal{C}_4| + \sum_{i\geq 5}(\frac{id}{2} + (i-1))|\mathcal{C}_i| - 3}{d-1} - 3|\mathcal{C}_4| - \sum_{i\geq 5}(i-1)|\mathcal{C}_i| \\
&\leq \frac{e(H) - 3 + (5-d)|\mathcal{C}_4| + \sum_{i \geq 5}\{\frac{id}{2} + (i-1) - (d-1)(i-1)\}|\mathcal{C}_i|}{d-1} \\
&= \frac{e(H) - 3 + (5-d)|\mathcal{C}_4| + \sum_{i\geq 5} \frac{(4-d)(i-2)+4}{2}|\mathcal{C}_i|}{d-1}.
\end{aligned} \tag{5}$$

We note that $d \geq 5$, so $(5-d)|C_4| \leq 0$. Clearly, if $d \geq 6$ or $\mathcal{C}_5 = \emptyset$, then we have $\sum_{i\geq 5} \frac{(4-d)(i-2)+4}{2}|\mathcal{C}_i| \leq 0$. Hence, if $d \geq 6$ or $\mathcal{C}_5 = \emptyset$, then $|\mathcal{S}| \leq (e(H)-3)/(d-1)$, and we are done. In order to deal with the remaining case $d = 5$ and $\mathcal{C}_5 \neq \emptyset$ we need to modify the operation for circuits in $\mathcal{C}_5$.

Let $C \in \mathcal{C}_5$. Then $H[V(C)] = C$ is a cycle on five vertices, because H is triangle-free. We write $C = u_1u_2u_3u_4u_5u_1$. Since $d = 5$, vertices with degree two or three cannot be adjacent to each other. Then C contains at least three vertices of degree at least four in H. By symmetry, we may assume that

$$d(u_1), d(u_3), d(u_4) \geq 4.$$

We modify the operation as follows. If $d(u_2), d(u_5) \geq 3$, we remove $E(C) \setminus \{u_1u_2\}$ and add three pendant edges to u_1 and four pendant edges to u_2, u_3, u_4, respectively, and we add five pendant edges to u_5. Then the net increase in the number of edges is 16. If one or both of u_2 and u_5 have degree 2, we add more pendant edges to (one of) them, but we can save some on their neighbors on C that now have degree at least five. One easily checks that in those cases we can get away with a net increase of 15 edges.

For every circuit in $\mathcal{C} \setminus \mathcal{C}_5$, we apply one of the previous operations as considered in Case 1 or Case 2. We can guarantee that the resulting graph H^* is a forest by Claim 2, and we can verify that it has minimum edge-degree $d = 5$. Then we can modify inequality (4) into

$$e(H^*) \leq e(H) + 12|\mathcal{C}_4| + 16|\mathcal{C}_5| + \sum_{i \geq 6}(\frac{5i}{2} + (i-1))|\mathcal{C}_i|,$$

and (5) into

$$\begin{aligned}|\mathcal{S}| &\leq \frac{e(H^*) - 3}{4} - 3|\mathcal{C}_4| - \sum_{i \geq 5}(i-1)|\mathcal{C}_i| \\ &\leq \frac{e(H) + 12|\mathcal{C}_4| + 16|\mathcal{C}_5| + \sum_{i \geq 6}(\frac{5l}{2} + (i-1))|\mathcal{C}_i| - 3}{4} - 3|\mathcal{C}_4| - \sum_{i \geq 5}(i-1)|\mathcal{C}_i| \\ &= \frac{e(H) - 3 + \sum_{i \geq 6}(-\frac{i}{2} + 3)|\mathcal{C}_i|}{4} \leq \frac{e(H) - 3}{4}.\end{aligned}$$

From the above discussions it follows that we can construct (in polynomial time) a dominating system that satisfies the upper bound of Theorem 8.