A simple combinatorial algorithm for restricted 2-matchings in subcubic graphs - via half-edges

Katarzyna Paluch † and Mateusz Wasylkiewicz ‡
Institute of Computer Science, University of Wrocław

Abstract
We consider three variants of the problem of finding a maximum weight restricted 2-matching in a subcubic graph $G$. (A 2-matching is any subset of the edges such that each vertex is incident to at most two of its edges.) Depending on the variant a restricted 2-matching means a 2-matching that is either triangle-free or square-free or both triangle- and square-free. While there exist polynomial time algorithms for the first two types of 2-matchings, they are quite complicated or use advanced methodology. For each of the three problems we present a simple reduction to the computation of a maximum weight $b$-matching. The reduction is conducted with the aid of half-edges. A half-edge of edge $e$ is, informally speaking, a half of $e$ containing exactly one of its endpoints. For a subset of triangles of $G$, we replace each edge of such a triangle with two half-edges. Two half-edges of one edge $e$ of weight $w(e)$ may get different weights, not necessarily equal to $\frac{1}{2}w(e)$.

In the metric setting when the edge weights satisfy the triangle inequality, this has a geometric interpretation connected to how an incircle partitions the edges of a triangle. Our algorithms are additionally faster than those known before. The running time of each of them is $O(n^2 \log n)$, where $n$ denotes the number of vertices in the graph.

1 Introduction

A subset $M$ of edges of an undirected simple graph is a 2-matching if every vertex is incident to at most two edges of $M$. 2-matchings belong to a wider class of $b$-matchings, where for every vertex $v$ in the set of vertices $V$ of the graph, we are given a natural number $b(v)$ and a subset of edges is a $b$-matching if every vertex is incident to at most $b(v)$ of its edges. A 2-matching is called $C_k$-free if it does not contain any cycle of length at most $k$. Note that every 2-matching is $C_2$-free and the smallest length of a cycle in a 2-matching is three. A 2-matching of maximum size can be found in polynomial time by a reduction to a classical matching. The $C_k$-free 2-matching problem consists in finding a $C_k$-free 2-matching of maximum size. Observe that the $C_k$-free 2-matching problem for $n/2 \leq k < n$, where $n$ is the number of vertices in the graph, is equivalent to finding a Hamiltonian cycle, and thus NP-hard. Hartvigsen [9] gave a complicated algorithm for the case of $k = 3$. Papadimitriou [5] showed that this problem is NP-hard when $k \geq 5$. The complexity of the $C_4$-free 2-matching problem is unknown.

In the weighted version of the problem, each edge $e$ is associated with a nonnegative weight $w(e)$ and we are interested in finding a $C_k$-free 2-matching of maximum weight, where the weight of a 2-matching $M$ is defined as the sum of weights of edges belonging to $M$. Vornberger [26] showed that the weighted $C_4$-free 2-matching problem is NP-hard. We refer to cycles of length three and four as triangles and squares, respectively.

*Partially supported by Polish National Science Center grant 2018/29/B/ST6/02633.
†abraka@cs.uni.wroc.pl
‡mateusz.wasylkiewicz@cs.uni.wroc.pl
In the paper we consider the following three problems in subcubic graphs: the weighted triangle-free 2-matching problem (i.e. the weighted $C_3$-free 2-matching problem), the weighted square-free 2-matching problem, in which we want to find a maximum weight 2-matching without any squares, but possibly containing triangles and the weighted $C_4$-free 2-matching problem. A graph is called cubic if its every vertex has degree 3 and is called subcubic if its every vertex has degree at most 3.

The weighted triangle-free 2-matching problem in subcubic graphs. The existing two polynomial time algorithms for this problem are the following. Hartvigsen and Li [11] gave a rather complicated primal-dual algorithm with running time $O(n^3)$ and a long analysis. The algorithm uses a type of so-called comb inequality. Kobayashi [15] devised a simpler algorithm using the theory of $M$-concave functions on finite constant-parity jump systems as well as makes $O(n^3)$ computations of a maximum weight $b$-matching for $b \in \{0, 1, 2\}^V$. Its running time is $O(n^3 \log n)$.

We present a simple combinatorial algorithm for the problem that uses one computation of a maximum weight $b$-matching for $b \in \{0, 1, 2\}^V$. Given a subcubic graph $G$, we replace some of its triangles with gadgets containing half-edges and define a function $b$ on the set of vertices in such a way that, any $b$-matching in the thus constructed graph $G'$ yields a triangle-free 2-matching. A half-edge of edge $e$ is, informally speaking, a half of $e$ containing exactly one of its endpoints. Half-edges have already been introduced in [20] and used in several subsequent papers. Here we use a different weight distribution among half-edges of one edge - two half-edges of one edge $e$ may be assigned different weights and not necessarily equal to $\frac{1}{2}w(e)$. In the metric setting when the edge weights satisfy the triangle inequality, this has a geometric interpretation connected to how an incircle partitions the edges of a triangle. The running time of our algorithm is $O(n^2 \log n)$. If the graph is unweighted, then the run time of this algorithm becomes $O(n^{3/2})$.

Square-free 2-matchings. In bipartite graphs a shortest cycle has length four - a square. Polynomial time algorithms for the $C_4$-free 2-matching problem in bipartite graphs were shown by Hartvigsen [10], Pap [21] and analyzed by Király [13]. As for the weighted version of the square-free 2-matching problem in bipartite graphs it was proven to be NP-hard [8][14] and solved by Makai [18] and Takazawa [23] for the case when the weights of edges are vertex-induced on every square of the graph. When it comes to the square-free 2-matching problem in general graphs, Nam [19] constructed a complex algorithm for it for graphs, in which all squares are vertex-disjoint. Bérczi and Kobayashi [3] showed that the weighted square-free 2-matching problem is NP-hard for general weights even if the given graph is cubic, bipartite and planar and gave a polynomial algorithm that finds a maximum weight 2-matching that contains no squares (but it can contain triangles). In [3] the square-free 2-matching problem is used for solving the $(n - 3)$-connectivity augmentation problem. As regards subcubic graphs, there are two other results besides those mentioned above. Bérczi and Végh [4] considered the problem of finding a maximum $t$-matching (a $b$-matching such that $b(v) = t$ for each vertex $v$) which does not contain any subgraph from a given set of forbidden $K_{t,t}$ and $K_{t+1}$ in an undirected graph of degree at most $t + 1$. Observe that the square-free 2-matching problem in subcubic graphs is a special case of this problem for $t = 2$.

The $C_4$-free 2-matching problem was previously investigated only in the unweighted version by Hartvigsen and Li in [11], who devised an $O(n^{3/2})$-algorithm. We present combinatorial algorithms for the weighted square-free 2-matching problem and the weighted $C_4$-free 2-matching problem for the case when the weights of edges are vertex-induced on every square of the graph and the graph is subcubic. These algorithms are similar to the one for the weighted triangle-free 2-matching problem in subcubic graphs and have the same running time.

Related work Some generalizations of the $C_4$-free 2-matching problem were investigated. Recently, Kobayashi [16] gave a polynomial algorithm for finding a maximum weight 2-matching that does not contain any triangle from a given set of forbidden edge-disjoint triangles. One can also consider non-simple $b$-matchings, in which every edge $e$ may occur in more than one copy. Problems
connected to non-simple $b$-matchings are usually easier than variants with simple $b$-matchings. Efficient algorithms for triangle-free non-simple 2-matchings (such 2-matchings may contain 2-cycles) were devised by Cornuéjols and Pulleyback [5,6], Babenko, Gusakov and Razenshtein [2], and Artamonov and Babenko [1]. Other results for restricted non-simple $b$-matchings appeared in [22,24,25].

2 Preliminaries

Let $G = (V, E)$ be an undirected graph with vertex set $V$ and edge set $E$. We denote the number of vertices of $G$ by $n$ and the number of edges of $G$ by $m$. We assume that all graphs are simple, i.e., they contain neither loops nor parallel edges. We denote an edge connecting vertices $v$ and $u$ by $(v, u)$. A cycle of graph $G$ is a sequence $c = (v_0, \ldots, v_{l-1})$ for some $l \geq 3$ of pairwise distinct vertices of $G$ such that $(v_i, v_{(i+1) \mod l}) \in E$ for every $i \in \{0, 1, \ldots, l-1\}$. We refer to $l$ as the length of $c$. For a given cycle $c = (v_0, \ldots, v_{l-1})$ any edge of $G$, which connects two vertices of $c$ and does not occur in $c$ is called a diagonal (of c). For a subgraph $H$ of $G$, we denote the edge set of $H$ by $E(H)$. For an edge set $F \subseteq E$ and $v \in V$, we denote by $\deg_F(v)$ the number of edges of $F$ incident to $v$.

An instance of each of the three problems that we consider in the paper consists of an undirected subcubic graph $G = (V, E)$ and a weight function $w : E \rightarrow \mathbb{R}_{\geq 0}$. In the weighted triangle-free 2-matching problem the goal is to find a maximum weight triangle-free 2-matching of $G$. In the weighted square-free (resp. $C_4$-free) 2-matching problem we additionally assume that the weights on the edges are vertex-induced on each square of $G$, i.e. for any square $s = (v_1, v_2, v_3, v_4)$ there exists a function $r : \{v_1, v_2, v_3, v_4\} \rightarrow R$ such that for any edge $e = (u, v)$ connecting two vertices of $s$ it holds that $w(e) = r(u) + r(v)$. The aim in the weighted square-free (resp. $C_4$-free) 2-matching problem is to compute a maximum weight square-free (corr. $C_4$-free) 2-matching of $G$.

We will use the classical notion of a $b$-matching, which is a generalization of a matching. For a vector $b \in \mathbb{N}^V$, an edge set $M \subseteq E$ is said to be a $b$-matching of $G$ if $\deg_M(v) \leq b(v)$ for every $v \in V$. Notice that a $b$-matching with $b(v) = 1$ for every $v \in V$ is a classical matching. A $b$-matching of $G$ of maximum weight can be computed in polynomial time. We refer to Lovász and Plummer [17] for further background on $b$-matchings.

We are interested in computing a $b$-matching of a graph $G$ where we are given vectors $l, u \in \mathbb{N}^V$ and a weight function $w : E \rightarrow \mathbb{R}$. For a vertex $v \in V$, $[l(v), u(v)]$ is said to be a capacity interval of $v$. An edge set $M \subseteq E$ is said to be an $(l, u)$-matching if $l(v) \leq \deg_M(v) \leq u(v)$ for every $v \in V$. An $(l, u)$-matching $M$ is said to be a maximum weight $(l, u)$-matching if there is no $(l, u)$-matching $M'$ of $G$ of weight greater than $w(M)$. A maximum weight $(l, u)$-matching can be computed efficiently.

**Theorem 1 ([7]).** There is an algorithm that, given a graph $G = (V, E)$, a weight function $w : E \rightarrow \mathbb{R}$ and vectors $l, u \in \mathbb{N}^V$, in time $O((\sum_{v \in V} u(v)) \min\{|E(G)| \log |V(G)|, |V(G)|^2\})$, finds a maximum weight $(l, u)$-matching of $G$.

Given an $(l, u)$-matching $M$ and an edge $e = (u, v) \in M$, we say that $u$ is matched to $v$ in $M$.

3 Outline of the Algorithm

The general scheme of the algorithm for each variant of the restricted 2-matching problem is the same - we give it below.
Algorithm 1 Computing a maximum weight restricted 2-matching of a subcubic graph $G$ given a weight function $w : E \rightarrow \mathbb{R}_{\geq 0}$.

Step 1. Construct an auxiliary graph $G' = (V', E')$ of size $O(n)$ by replacing some triangles and/or squares of $G$ with gadgets containing half-edges. (Both gadgets and half-edges are defined later.)

Step 2. Define a weight function $w' : E' \rightarrow \mathbb{R}$ and vectors $l, u \in \mathbb{N}^{V'}$ such that $u(v) \leq 2$ for every $v \in V'$.

Step 3. Compute a maximum weight $(l, u)$-matching $M'$ of $G'$.

Step 4. Construct a 2-matching $M$ of $G$ by replacing all half-edges of $M'$ with some edges of $G$ in such a way that $w(M) \geq w'(M')$.

Step 5. Remove the remaining triangles and/or squares from $M$ by replacing some of their edges with other ones without decreasing the weight of $M$.

Claim 1. Algorithm 1 runs in time $O(n^2 \log n)$.

Proof. It will be easy to implement all steps of Algorithm 1 except Step 3 in linear time. Hence, the running time of our algorithm is equal to the running time of an algorithm for computing a maximum weight $(l, u)$-matching of $G'$, i.e., it is equal to $O((\sum_{v \in V'} u(v)) \min\{|E'| \log |V'|, |V'|^2\})$. Recall that $|V'| + |E'| = O(n)$ and $u(v) \leq 2$ for every $v \in V'$. Hence, the running time of Step 3 is $O(n^2 \log n)$.

Let us also remark that in the unweighted versions of the problem Algorithm 1 runs in $O(n^{3/2})$.

4 Triangle-free 2-matchings in subcubic graphs

In this section we solve a maximum weight triangle-free 2-matching problem in subcubic graphs. We assume that each connected component of $G$ is different from $K_4$, i.e., different from a 4-vertex clique.

One can observe that, since $G$ is subcubic, any edge $e$ of $G$ belongs to at most two different triangles. Also, any triangle of $G$ shares an edge with at most one other triangle or, in other words, any triangle of $G$ is not edge-disjoint with at most one other triangle.

Definition 1. A triangle $t$, which has a common edge with some other triangle $t'$ such that $w(t) \leq w(t')$ is said to be unproblematic. Otherwise, $t$ is said to be problematic.

Unproblematic triangles can be easily got rid of from any 2-matching $M$ of $G$ by replacing some of its edges with other ones as explained in more detail in the proof of Theorem 1.

Observe that any problematic triangle of $G$ is vertex-disjoint with any other problematic triangle of $G$.

We begin with the following simple fact.

Claim 2. Let $t = (a, b, c)$ be a triangle of $G$, whose edges have weights $w(a, b), w(b, c), w(c, a)$, respectively. Then, there exist real numbers $r_a, r_b, r_c$ such that $w(a, b) = r_a + r_b$, $w(b, c) = r_b + r_c$ and $w(c, a) = r_c + r_a$.

If the weights of edges of $t$ satisfy the triangle inequality, then Claim 2 has a geometric interpretation connected to how an incircle partitions the edges of a triangle - see Figure 1.
If $G$ contains at least one problematic triangle, we build a graph $G' = (V', E')$ together with a weight function $w' : E' \to \mathbb{R}$, in which each problematic triangle $t$ is replaced with a subgraph, called a **gadget for $t$**. The precise construction of $G'$ is the following. We start off with $G$.

Let $t = (a, b, c)$ be any problematic triangle of $G$. For each edge $(p, q)$ of $t$ we add two new vertices $v_{pq}$ and $v_{qp}$, called **subdivision vertices** (of $t$), and we replace $(p, q)$ with three new edges: $(p, v_{pq}), (v_{pq}, v_{qp}), (v_{qp}, q)$. Each of the edges $(p, v_{pq}), (v_{pq}, v_{qp})$ is called a **half-edge** (of $(p, q)$ and also of $t$). The edge $(v_{pq}, v_{qp})$ is called an **eliminator** (of $(p, q)$). The half-edges of edges of $t$ get weights equal to values of $r_a, r_b, r_c$ from Claim 2, i.e., $w'(a, v_{ab}) = w'(a, v_{ac}) = r_a, w'(b, v_{ba}) = w'(b, v_{bc}) = r_b$ and $w'(c, v_{ca}) = w'(c, v_{cb}) = r_c$. The weight of each eliminator is 0. Additionally, we introduce four new vertices $u_a, u_b, u_c, u_t$, called **global vertices**. For every $d \in \{a, b, c\}$ we connect $u_d$ with $u_t$ and with every subdivision vertex connected to $d$. Every edge incident to a global vertex has weight 0.

We define vectors $l, u \in \mathbb{N}^{V'}$ as follows. We set a capacity interval of every vertex of the original graph $G$ to $[0, 2]$ and we set a capacity interval of every other vertex of $G'$ to $[1, 1]$, i.e., every vertex of $V' \setminus V$ is matched to exactly one vertex of $G'$ in any $(l, u)$-matching of $G'$.

The main ideas behind the gadget for a problematic triangle $t = (a, b, c)$ are the following. An $(l, u)$-matching $M'$ of $G'$ is to represent roughly a triangle-free 2-matching $M$ of $G$. If $M'$ contains both half-edges of some edge $e$, then $e$ is included in $M$. If $M'$ contains an eliminator of $e$, then $e$ does not belong to $M$ (is excluded from $M$). We want to ensure that at least one edge of $t$ does not belong to $M$. This is done by requiring that two of the global vertices $u_a, u_b, u_c$ are matched to subdivision vertices. In this way two half-edges of $t$ are guaranteed not to belong to $M'$ and hence to $M$.  

![Figure 1: Partition of the edges of a triangle by its incircle.](image1)

![Figure 2: A gadget for a problematic triangle $t = (a, b, c)$.](image2)
(l, u)-matchings of $G'$.

**Theorem 2.** Let $M$ be any triangle-free 2-matching of $G$. Then we can find an $(l, u)$-matching $M'$ of $G'$ such that $w'(M') = w(M)$.

**Proof.** We initialize $M'$ as the empty set. We add every edge of $M$ that does not belong to any problematic triangle of $G$ to $M'$. Consider any problematic triangle $t = (a, b, c)$ of $G$. Since $M$ is triangle-free, there exists an edge of $t$ that does not belong to $M$. If more than one edge of $t$ does not belong to $M$, we choose one of them. Suppose that we chose $(a, b) \notin M$. Then we add edges $(v_a^b, u_a)$, $(v_a^b, u_b)$ and $(u_t, u_c)$ to $M'$. For every other edge $e$ of $t$ we proceed as follows. If $e \in M$, we add both half-edges of $e$ to $M'$, otherwise we add the eliminator of $e$ to $M'$. Since the weight of any edge of $t$ in $G$ is equal to the sum of the weights of its half-edges in $G'$, we get that $w'(M') = w(M)$.

**Theorem 3.** Let $M'$ be any $(l, u)$-matching of $G'$. Then we can find a triangle-free 2-matching $M$ of $G$ such that $w(M) \geq w'(M')$.

**Proof.** We initialize $M$ as the empty set. We add every edge of $M'$ that belongs to $G$ to $M$. For every problematic triangle of $G$ we will add some of its edges to $M$.

Consider any problematic triangle $t = (a, b, c)$ of $G$. Notice that exactly two of the vertices $u_a, u_b, u_c$ are matched to subdivision vertices, because $u_t$ is matched to one of $u_a, u_b, u_c$. This corresponds to excluding two half-edges of $t$ from $M$. Since every subdivision vertex is required to be matched to exactly one vertex in $G'$, we get that an even number and at most four subdivision vertices of $t$ are matched to the vertices $a, b, c$. This indicates, which half-edges of $t$ are going to be included in $M$. Observe that the two subdivision vertices that are matched in $M'$ to vertices $u_a, u_b, u_c$ are adjacent to two different vertices of $t$. Thus, we have:

**Claim 3.** If $M'$ contains exactly four half-edges of $t$, then the two half-edges of $t$ that do not belong to $M'$ are not adjacent to the same vertex of $t$.

Every other subset of half-edges of $t$ containing an even number of at most four half-edges of $t$ can occur in $M'$.

In each of these cases, we proceed as follows (see Figure 3):

1. Exactly zero subdivision vertices of $t$ are matched to $a, b, c$. We do not include any edge of $t$ in $M$.

2. Exactly two subdivision vertices of $t$ are matched to $a, b, c$.

   (a) The two subdivision vertices of $t$ are matched to two different vertices $u, v$ of $t$. Then we include the edge $(u, v)$ in $M$.

   (b) The two subdivision vertices of $t$ are matched to the same vertex $u$ of $t$. Then we include in $M$ two edges of $t$ incident to $u$. (This is the only case where $w(M)$ may be greater than $w'(M')$ when it comes to half-edges of $t$. Notice that for any two vertices $u, v$ of $t$ we have that $r_u + r_v \geq 0$).

3. Exactly four subdivision vertices of $t$ are matched to $a, b, c$. Then, by Claim 3, two of these vertices are matched to the same vertex $u$ of $t$ and the other two are matched to the remaining two vertices of $t$. In this case we include in $M$ two edges of $t$ incident to $u$.

Since half-edges incident to the same vertex have the same weight, we get that $w(M) \geq w'(M')$.

The resulting 2-matching $M$ can contain some unproblematic triangles. We remove them one by one. Let $t = (a, b, c)$ be any such triangle. From Definition 1 there exists another triangle $t' =
(a, b, d), which shares an edge with t and such that \( w(t') \geq w(t) \). Hence, either \( w(a, d) \geq w(a, c) \) or \( w(b, d) \geq w(b, c) \). Assume that \( w(a, d) \geq w(a, c) \). We replace the edge \( (a, c) \) with the edge \( (a, d) \) without decreasing the weight of \( M \). □

![Figure 3: The construction of a maximum weight triangle-free 2-matching of \( G \) from a maximum weight \((l, u)\)-matching of \( G' \).](image)

### 5 Square-free 2-matchings in subcubic graphs

In this section we solve a maximum weight square-free 2-matching problem in subcubic graphs. Recall that this problem is NP-hard for general weights, therefore we assume that weights are vertex-induced on every square, i.e., for any square \( s = (a, b, c, d) \) of \( G \) there exist real numbers \( r_a, r_b, r_c, r_d \), called potentials of \( s \) such that for any edge \( e = (u, v) \) connecting two vertices of \( s \) it holds that \( w(e) = r_u + r_v \). (Note that if a given edge \( e = (u, v) \) belongs to two different squares \( s \) and \( s' \), then potentials of \( s \) and \( s' \) on \( u \) and \( v \) may be different.)

We also assume that each connected component of \( G \) is different from \( K_4 \).

For a square \( s = (v_0, v_1, v_2, v_3) \) of \( G \), edges \((v_0, v_1), (v_1, v_2), (v_2, v_3)\) and \((v_3, v_0)\) are said to be native edges of \( s \).

One can observe that, since \( G \) is subcubic, any two different squares of \( G \) are vertex-disjoint or have either one or two edges in common.

**Definition 2.** A square \( s \) of \( G \) is said to be **unproblematic** if there exists another square \( s' \) such that (i) \( s \) shares exactly one edge with \( s' \) or (ii) \( s \) shares two edges with \( s' \) and \( w(s) \leq w(s') \). Otherwise, \( s \) is said to be **problematic**.

Observe that any problematic square of \( G \) is vertex-disjoint with any other problematic square of \( G \).

The following simple observation shows that squares which have exactly one common edge with another square do not pose any problem for computing a maximum weight square-free 2-matching of \( G \).

**Claim 4.** Consider any two squares \( s = (a, b, c, d) \) and \( s' = (c, d, e, f) \) of \( G \) which share exactly one edge. Let \( M_1 \) be a 2-matching of \( G \) that contains \( s \). Then there exists a 2-matching \( M_2 \) of \( G \), which does not contain \( s \) or any square not already contained in \( M_1 \) and such that \( w(M_2) \geq w(M_1) \).

**Proof.** We set \( M_2 = M_1 \setminus \{(c, d), (e, f)\} \cup \{(c, f), (d, e)\} \). Note that we can assume that \( M_1 \) contains the edge \((e, f)\), because \( G \) is subcubic and \( M_1 \) contains neither \((c, f)\) nor \((d, e)\). It is straightforward to check that \( M_2 \) is a 2-matching of \( G \) that does not contain \( s \). Furthermore, the given construction
does not introduce any additional squares into $M_2$. Observe that $w(M_2) \geq w(M_1)$, since $w$ is vertex-induced on $s'$.

We show the construction of a gadget for a problematic square $s = (a, b, c, d)$. We use the notation introduced in Section 4. For every native edge $(p, q)$ of $s$, we introduce two subdivision vertices $v^p_q, v^q_p$ and replace $(p, q)$ with two half-edges $(p, v^p_q)$ and $(v^q_p, q)$ and an eliminator $(v^p_q, v^q_p)$. (We do not replace any diagonal of $s$.) Additionally, we introduce two new global vertices $u^1_s$ and $u^2_s$. We connect $u^1_s$ with all subdivision vertices adjacent to either $a$ or $c$. Symmetrically, we connect $u^2_s$ with all subdivision vertices adjacent to either $b$ or $d$. The half-edges incident to $a, b, c$ and $d$ get weight $r_a, r_b, r_c$ and $r_d$, respectively, where $r_a, \ldots, r_d$ are potentials of $s$. All other edges of the gadget get weight 0. We set a capacity interval of every vertex of $s$ to $[0, 2]$ and we set a capacity interval of every other vertex of the gadget to $[1, 1]$.

![Figure 4: A gadget for a problematic square $s = (a, b, c, d)$.](image)

**Theorem 4.** Let $M$ be any square-free 2-matching of $G$. Then we can find an $(l, u)$-matching $M'$ of $G'$ such that $w(M') = w(M)$.

**Proof.** We initialize $M'$ as the empty set. We add every edge of $M$ that does not belong to any problematic square of $G$ to $M'$.

Consider any problematic square $s = (a, b, c, d)$ of $G$. Assume that $(a, b)$ does not belong to $M$. We add edges $(v^a_d, u^1_s)$ and $(v^b_d, u^2_s)$ to $M'$. For every other native edge $e$ of $s$ we proceed as follows. If $e \in M$, we add both half-edges of $e$ to $M'$, otherwise we add the eliminator of $e$ to $M'$.

**Theorem 5.** Let $M'$ be any $(l, u)$-matching of $G'$. Then we can find a square-free 2-matching $M$ of $G$ such that $w(M) \geq w'(M')$.

**Proof.** We initialize $M$ as the empty set. We add every edge of $M'$ that belongs to $G$ to $M$. For every problematic square of $G$ we will add some of its edges to $M$. Next we will replace some edges of $M$ with other ones to remove unproblematic squares.

Consider any problematic square $s = (a, b, c, d)$ of $G$. Notice that there exists a native edge $(p, q)$ of $s$ such that $u^1_p$ and $u^2_q$ are matched in $M'$ to two subdivision vertices, one of which is adjacent to $p$ and the other to $q$. W.l.o.g. assume that $(p, q) = (a, b)$. We consider the following cases:
1. $u_1$ and $u_2$ are matched in $M'$ to $v_a$ and $v_b$, respectively. We add every native edge of $s$ whose both half-edges belong to $M'$ to $M$. Notice that for every other native edge $e$ of $s$, the eliminator of $e$ belongs to $M'$.

2. Either $u_1$ is matched to $v_a$ or $u_2$ is matched to $v_b$ in $M'$, but not both of them. Assume that $u_1$ is matched to $v_a$. Therefore, edges $(u_2, v_a)$ and $(b, v_b)$ belong to $M'$. We replace these two edges with $(u_2, v_a)$ and $(b, v_b)$ without changing the weight of $M'$. Then we proceed as in case 1.

3. $u_1$ and $u_2$ are matched to $v_a$ and $v_b$, respectively, in $M'$. If $(v_a, v_b)$ does not belong to $M'$, we connect $u_1$ and $u_2$ with $(v_a, v_b)$, respectively, similarly as in case 2 and we proceed as in case 1. Assume now that $(v_a, v_b)$ belongs to $M'$. Notice that $(d, v_d)$ and $(c, v_c)$ belong to $M'$. We add $(a, d)$ and $(b, c)$ to $M$. Additionally, if both half-edges of $(c, d)$ belong to $M'$, we add $(c, d)$ to $M$.

The resulting 2-matching $M$ can contain some unproblematic squares. We remove squares, which share exactly one edge with another square from $M$ one by one using Claim 4. We remove the rest of unproblematic squares in a similar way as we got rid of unproblematic triangles in the proof of Theorem 3. Each such removal does not introduce any squares into $M$, therefore $M$ is a square-free 2-matching in the end. □

6 $C_4$-free 2-matchings in subcubic graphs

In this section we solve a maximum weight $C_4$-free 2-matching problem in subcubic graphs. We assume that weights are vertex-induced on every square. We also assume that each connected component of $G$ is different from $K_4$.

We say that a cycle $C$ of $G$ is short if it is either a triangle or a square. We say that a short cycle $C$ of $G$ is unproblematic if it shares exactly one edge with some square of $G$ or if it fits Definition 1 or Definition 2. A short cycle, which is not unproblematic is said to be problematic.

We have the analogue of Claim 4, which justifies considering triangles sharing one edge with a square unproblematic:

Claim 5. Consider two short cycles: a triangle $t = (a, c, d)$ and a square $s' = (c, d, e, f)$ of $G$ which share exactly one edge. Let $M_1$ be a 2-matching of $G$ that contains $t$. Then there exists a 2-matching $M_2$ of $G$, which does not contain $t$ or any short cycle not already contained in $M_1$ and such that $w(M_2) \geq w(M_1)$.

Observe that any two different short problematic cycles that are not vertex-disjoint must form a pair consisting of a square $s = (a, c, b, d)$ and a triangle $t_1 = (a, c, d)$ with exactly two common edges. We call a subgraph induced on vertices of such $s$ and $t_1$ a double triangle $T = (a, b, c, d)$. In $G'$ we build the following gadget for every double triangle.

Consider any double triangle $T = (a, b, c, d)$. We remove $c$ and $d$ from $G'$ and we add a vertex $u_T$ to $G'$. We connect $v_1^T$ with $v_2^T$ and we connect $u_T$ with both $a$ and $b$. Let $M_1^T(T)$ denote the weight of a maximum weight $C_4$-free 2-matching of $T$ in which $a$ has degree $i$ and $b$ has degree $j$. We set the weight of edges $(v_1^T, v_2^T)$, $(u_T, a)$ and $(u_T, b)$ to $M_1^T(T)$, $M_2^T(T) - M_1^T(T)$ and $M_2^T(T)$, respectively. We set capacity intervals of $a$ and $u_T$ to $[0, 1]$. We set capacity intervals of $v_1^T$ and $v_2^T$ to $[1, 1]$. For every problematic short cycle that is not part of any double triangle we add a corresponding gadget presented in Section 4 or Section 5.

Theorem 6. Let $M$ be any $C_4$-free 2-matching of $G$. Then we can find an $(l, u)$-matching $M'$ of $G'$ such that $w(M') \geq w(M)$.
Proof. We initialize $M'$ as the empty set. We add to $M'$ every edge of $M$ that belongs to no problematic short cycle.

Consider any double triangle $T = (a, b, c, d)$ of $G$. We add $(v^1_T, v^2_T)$ to $M'$. Let $\hat{M} = M \cap E(T)$. If $\deg_{\hat{M}}(a) = 2$, then we add $(u_T, a)$ to $M'$. If $\deg_{\hat{M}}(b) = 2$, then we add $(u_T, b)$ to $M'$. Note that $\deg_{\hat{M}}(a) \leq 1$ or $\deg_{\hat{M}}(b) \leq 1$, therefore $\deg_{M'}(u_T) \leq 1$.

For every problematic short cycle that is not part of any double triangle we add edges of a corresponding gadget to $M'$ in the same way as we did in the proofs of Theorem 2 and Theorem 4.

Theorem 7. Let $M'$ be any $(l, u)$-matching of $G'$. Then we can find a $C_4$-free 2-matching $M$ of $G$ such that $w(M) \geq w'(M')$.

Proof. We initialize $M$ as the empty set. We add to $M$ every edge of $M'$ that belongs to $G$.

Consider any double triangle $T = (a, b, c, d)$ of $G$. Let $i$ and $j$ denote the number of edges of the gadget for $T$ incident to $a$ and $b$, respectively. Notice that $i + j \leq 1$. We add to $M$ a maximum weight $C_4$-free 2-matching of $T$ in which $a$ has degree $i + 1$ and $b$ has degree $j + 1$.

For every short cycle that is not part of any double triangle we proceed in the same way as in the proofs of Theorem 3 and Theorem 5.

References

[1] S. Artamonov and M. Babenko. A fast scaling algorithm for the weighted triangle-free 2-matching problem. European Journal of Combinatorics, 68:3–23, 2018.

[2] M. Babenko, A. Gusakov, and I. Razenshteyn. Triangle-free 2-matchings revisited. In Computing and Combinatorics, pages 120–129, 2010.

[3] K. Bérczi and Y. Kobayashi. An algorithm for $(n-3)$-connectivity augmentation problem: Jump system approach. Journal of Combinatorial Theory, Series B, 102(3):565–587, 2012.

[4] K. Bérczi and L. Végh. Restricted b-matchings in degree-bounded graphs. In Integer Programming and Combinatorial Optimization, pages 43–56, 2010.

[5] G. Cornuéjols and W. Pulleyblank. A matching problem with side conditions. Discrete Mathematics, 29(2):135–159, 1980.

[6] G. Cornuéjols and W. Pulleyblank. Perfect triangle-free 2-matchings. In Combinatorial Optimization II, Mathematical Programming Studies, pages 1–7. Springer Berlin Heidelberg, 1980.
[7] H. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing, pages 448–456, 1983.

[8] J. Geelen. The $C_6$-free 2-factor problem in bipartite graphs is NP-complete. Unpublished, 1999.

[9] D. Hartvigsen. Extensions of Matching Theory. PhD thesis, Carnegie-Mellon University, 1984.

[10] D. Hartvigsen. Finding maximum square-free 2-matchings in bipartite graphs. Journal of Combinatorial Theory, Series B, 96(5):693–705, 2006.

[11] D. Hartvigsen and Y. Li. Maximum cardinality simple 2-matchings in subcubic graphs. SIAM Journal on Optimization, 21(3):1027–1045, 2011.

[12] D. Hartvigsen and Y. Li. Polyhedron of triangle-free simple 2-matchings in subcubic graphs. Mathematical Programming, 138:43–82, 2013.

[13] Z. Király. $C_4$-free 2-factors in bipartite graphs. Technical report, Egerváry Research Group, 1999.

[14] Z. Király. Restricted $t$-matchings in bipartite graphs. Technical report, Egerváry Research Group, 2009.

[15] Y. Kobayashi. A simple algorithm for finding a maximum triangle-free 2-matching in subcubic graphs. Discrete Optimization, 7:197–202, 2010.

[16] Y. Kobayashi. Weighted triangle-free 2-matching problem with edge-disjoint forbidden triangles. In Integer Programming and Combinatorial Optimization, pages 280–293, 2020.

[17] L. Lovász and M. Plummer. Matching theory. AMS Chelsea Publishing, corrected reprint of the 1986 original edition, 2009.

[18] M. Makai. On maximum cost $K_{t,t}$-free $t$-matchings of bipartite graphs. SIAM Journal on Discrete Mathematics, 21:349–360, 2007.

[19] Y. Nam. Matching Theory: Subgraphs with Degree Constraints and other Properties. PhD thesis, University of British Columbia, 1994.

[20] K. Paluch, K. Elbassioni, and A. van Zuylen. Simpler approximation of the maximum asymmetric traveling salesman problem. In 29th International Symposium on Theoretical Aspects of Computer Science, pages 501–506, 2012.

[21] G. Pap. Combinatorial algorithms for matchings, even factors and square-free 2-factors. Mathematical Programming, 110:57–69, 2007.

[22] G. Pap. Weighted restricted 2-matching. Mathematical Programming, 119:305–329, 2009.

[23] K. Takazawa. A weighted $K_{t,t}$-free $t$-factor algorithm for bipartite graphs. Mathematics of Operations Research, 34(2):351–362, 2009.

[24] K. Takazawa. Excluded $t$-factors in bipartite graphs: A unified framework for nonbipartite matchings and restricted 2-matchings. In Integer Programming and Combinatorial Optimization, pages 430–441, 2017.
[25] K. Takazawa. Finding a maximum 2-matching excluding prescribed cycles in bipartite graphs. *Discrete Optimization*, 26:26–40, 2017.

[26] O. Vornberger. Easy and hard cycle covers. Technical report, Universität Paderborn, 1980.