Opuscula Math. 45, no. 1 (2025), 79–101
https://doi.org/10.7494/OpMath.2025.45.1.79 Opuscula Mathematica

ON UNIQUENESS OF PACKING OF THREE COPIES OF 2-FACTORS

Igor Grzelec, Tomáš Madaras, and Alfréd Onderko

Communicated by Andrzej Żak

Abstract. The packing of three copies of a graph G is the union of three edge-disjoint copies (with the same vertex set) of G. In this paper, we completely solve the problem of the uniqueness of packing of three copies of 2-regular graphs. In particular, we show that C_3, C_4, C_5, C_6 and $2C_3$ have no packing of three copies, $C_7, C_8, C_3 \cup C_4, C_4 \cup C_4, C_3 \cup C_5$ and $3C_3$ have unique packing, and any other collection of cycles has at least two distinct packings.

Keywords: uniquely packable graph, 2-factor, 3-packing.

Mathematics Subject Classification: 05C70.

1. INTRODUCTION

All graphs considered in this paper are finite, undirected and have neither loops nor multiple edges. For a graph G, we will denote its order $|V(G)|$ and size $|E(G)|$ as n and m, respectively.

At the beginning, we present additional definitions which will be useful to formulate the results. For two graphs G_1 and G_2 with disjoint vertex sets, the *union* $G = G_1 \cup G_2$ has $V(G) = V(G_1) \cup V(G_2)$ and $E(G) = E(G_1) \cup E(G_2)$. The union of $n \geq 2$ disjoint copies of a graph H is denoted by $G = nH$. Further, for graphs G_1 and G_2 such that $V(G_1) = V(G_2)$ and $E(G_1) \cap E(G_2) = \emptyset$, the *edge-sum* $G_1 \oplus G_2$ has $V(G) = V(G_1) = V(G_2)$ and $E(G) = E(G_1) \cup E(G_2)$.

Let G be a graph of order n. An *embedding* of G into complete graph K_n is an injective mapping $\phi : V(G) \to V(K_n)$ such that $\phi(x)\phi(y) \in E(K_n)$ whenever $xy \in E(G)$. Denote by $\phi(G)$ the graph with the vertex set V and the edge set $\phi^*(E)$ where the mapping ϕ^* is induced by ϕ. A *packing* of l graphs $G_1, G_2, \ldots, G_l$ ($l \geq 2$) into K_n is a l-tuple $\Phi = (\phi_1, \phi_2, \ldots, \phi_l)$ such that, for each $i = 1, 2, \ldots, l$, ϕ_i is an embedding of G_i into K_n and $\phi_i(G_i)$ and $\phi_j(G_j)$ are edge-disjoint subgraphs of K_n for any $i \neq j$. When all G_i are isomorphic to G, Φ is called a *l-packing* of G.

One of the first results on packing problem is the following theorem, which was proved independently in [3, 4] and [15]:

Theorem 1.1. *Let $G = (V, E)$ be a graph of order n and size m. If $m \leq n - 2$, then G has a 2-packing.*

We can easily see that the star $K_{1,n-1}$ has not a 2-packing. Therefore, Theorem 1.1 cannot be improved by raising the size of G. Burns and Schuster in [5] described the full characterization of graphs of order n and size $n - 1$ that have a 2-packing:

Theorem 1.2. *Let $G = (V, E)$ be a graph of order n and size m. If $m \leq n - 1$, then either G has a 2-packing or G is isomorphic to one of the following graphs: $K_{1,n-1}$, $K_{1,n-4} \cup K_3$ with $n \geq 8$, $K_1 \cup K_3$, $K_2 \cup K_3$, $K_1 \cup 2K_3$, $K_1 \cup C_4$.*

Considering the problem of the uniqueness of graph packings, let us explain first what we mean by unique l-packings. From the definition of the l-packing of G, we may create the graph $\tilde{\Phi}(G) = \phi_1(G) \oplus \phi_2(G) \oplus \ldots \oplus \phi_l(G)$. Let $\Phi' = (\phi'_1, \phi'_2, \ldots, \phi'_l)$ be another l-packing of G. If $\tilde{\Phi}(G)$ and $\tilde{\Phi}'(G)$ are isomorphic, then we call Φ and Φ' isomorphic. We say that the l-packing of G is unique if all l-packings of G are isomorphic, otherwise G is not *uniquely l-packable.*

The problem of the uniqueness of 2-packing of graphs has been the subject of three papers. The next theorem from [19] characterizes all graphs of order n and size $n - 2$ that are uniquely 2-packable:

Theorem 1.3. *Let G be a graph of order n and size $m = n - 2$. Then either G is not uniquely 2-packable or G is isomorphic to one of the six following graphs: $K_2 \cup K_1$, $2K_2$, $K_3 \cup 2K_1$, $K_3 \cup K_2 \cup K_1$, $K_3 \cup 2K_2$, $2K_3 \cup 2K_1$.*

The following characterization of uniquely 2-packable forests was proved in [14].

Theorem 1.4. *Let F be a forest of order n having at least one edge. Then either F is not uniquely 2-packable or F is isomorphic to one of the following graphs: $K_2 \cup K_1$, $2K_2$, $3K_2$, the double star $S(p, q)$ or the $(n - 1)$-vertex star with one edge subdivided.*

Recently the problem of the uniqueness of 2-packings was completely solved for 2-factors, i.e. a vertex-disjoint union of cycles. More precisely, the following theorem was proved in [9]:

Theorem 1.5. *Let G be a vertex-disjoint union of k cycles. If G is C_3, C_4 or $2C_3$ then G is not 2-packable. The graphs C_5, C_6, $C_3 \cup C_4$, $C_3 \cup C_5$, $3C_3$ and $4C_3$ are uniquely 2-packable. In every other case, there exist at least two distinct 2-packings of G.*

For other results on different packing problems, we refer the reader to the survey papers [18, 20, 22].

It is worth to mention two results on the existence of 3-packing of a graph. In [21], the following theorem (which yields full characterization of all graphs of order n and size $n - 2$ that have a 3-packing) was proved:

Theorem 1.6. *Let G be a graph of order n and size m. If $m \leq n - 2$, then either there exists a 3-packing of G or $G \in \{K_3 \cup 2K_1, K_4 \cup 4K_1\}$.*

In [16], the full characterization of all trees that have a 3-packing was proved:

Theorem 1.7. *Let T be a tree of order $n \geq 6$ which is neither a star nor a star with one edge subdivided, nor else a 6-vertex star with one edge subdivided twice. Then there exists a 3-packing of T.*

The purpose of this paper is to consider the problem of the uniqueness of 3-packing of graphs for 2-factors. This problem for 2-factors is related to the well-known Oberwolfach problem which is still open. To present this problem in a formal way, we provide additional definitions. By a 2-*factorization* of a graph G we mean an edge-disjoint partition of the edge set of G into 2-factors. A 1-factor of a graph G (i.e. a perfect matching of G) will be denoted by I. The Oberwolfach problem (OP for short) asks whether a complete graph K_n (for n odd), or K_n without a 1-factor (for n even), admits a 2-factorization in which each 2-factor is isomorphic to a given 2-factor. More precisely, an instance $\mathrm{OP}(n; n_1, \ldots, n_k)$ of the Oberwolfach problem asks if there is a 2-factorization of K_n for n odd, or $K_n \setminus I$ for n even, such that each 2-factor is isomorphic to $C_{n_1} \cup C_{n_2} \cup \ldots \cup C_{n_k}$. Since the problem was posed in 1967 by Gerhard Ringel, many papers on the topic were published. With an exception of four cases, namely $\mathrm{OP}(6; 3^2)$, $\mathrm{OP}(9; 4, 5)$, $\mathrm{OP}(11; 3^2, 5)$ and $\mathrm{OP}(12; 3^4)$ (here, the superscript refers to the repetition of the number), for which solutions do not exist, solutions were obtained for all orders $n \leq 40$ (see [1] and [8]) and for many special cases (for example $\mathrm{OP}(n; r^k, n - rk)$ for all $n \geq 6kr - 1$, see [12]). For more results on this topic, we refer the reader to the survey [6].

Now, we introduce the relation between the problem of the uniqueness of 3-packing of a 2-factor and the Oberwolfach problem. Because the sum of three edge disjoint copies (with common vertex set) of any 2-factor is a 6-regular graph, 2-factors for which there exists a 3-packing shall have order at least seven.

Observation 1.8. *For any 2-factor $G = C_{n_1} \cup C_{n_2} \cup \ldots \cup C_{n_k}$ of order $n = n_1 + \ldots + n_k$, where $n \geq 7$, if there exists a solution for the instance $OP(n; n_1, \ldots, n_k)$ of the Oberwolfach problem then there exists a 3-packing of G.*

Remark that the converse of Observation 1.8 does not hold; it suffices to find a 3-packing of $C_4 \cup C_5$, because $\mathrm{OP}(9; 4, 5)$ has no solution. The 3-packing of this 2-factor is presented in Section 5. Moreover, considering complements of 3-packings of 2-factor of order seven and eight, we can easily see that they are isomorphic to a graph of size zero and order seven, and to a perfect matching of order eight, respectively. Therefore, the following observation holds:

Observation 1.9. *For any 2-factor $G = C_{n_1} \cup C_{n_2} \cup \ldots \cup C_{n_k}$ of order $n = n_1 + \ldots + n_k$, where $n \in \{7, 8\}$, if there exists a solution for the instance $OP(n; n_1, \ldots, n_k)$ of the Oberwolfach problem, then there exist a unique 3-packing of G.*

Note that the above observations can be generalized to the problem of the uniqueness of l-packing, for $l \geq 4$, of 2-factors using similar reasoning. For any 2-factor $G = C_{n_1} \cup C_{n_2} \cup \ldots \cup C_{n_k}$ of order $n = n_1 + \ldots + n_k$ with $n \geq 2l + 1$ such that

$\mathrm{OP}(n; n_1, \ldots, n_k)$ has a solution, it is easy to see that there exists l-packing of G. Furthermore, for any 2-factor $G = C_{n_1} \cup C_{n_2} \cup \ldots \cup C_{n_k}$ of order $n = n_1 + \ldots + n_k$, where $n \in \{2l+1, 2l+2\}$, if there exists a solution for $\mathrm{OP}(n; n_1, \ldots, n_k)$, then there exists a unique l-packing of G.

Now, we introduce our main result.

Theorem 1.10. *Let $G = C_{n_1} \cup C_{n_2} \cup \ldots \cup C_{n_k}$ be a vertex-disjoint union of k cycles. For cycles C_3, C_4, C_5, C_6 and the graph $2C_3$, there is no 3-packing. The graphs C_7, C_8, $C_3 \cup C_4$, $C_4 \cup C_4$, $C_3 \cup C_5$ and $3C_3$ have unique 3-packing. For any other graph G, there exist at least two distinct 3-packings.*

The proof of Theorem 1.10 is presented in the next sections. Section 2 contains the case of cycles, Section 3 presents the proof of the existence of 3-packing of 2-factors (for $k \geq 2$) and general strategy of the remaining part of the proof, Section 4 contains the proof of the existence of at least two distinct 3-packings of 2-factors for five particular families of 2-factors, and the last section presents the proof for the remaining small cases.

Remark. To better differentiate between copies of G in a 3-packing $\Phi(G) = (\phi_1, \phi_2, \phi_3)$ (both in subsequent figures and proofs), we say that $\phi_1(G)$ (the first or initial copy of G) is *black*, $\phi_2(G)$ (the second copy of G) is *red* and $\phi_3(G)$ (the third copy of G) is *blue* in such a 3-packing of G; this is useful, in particular, when a 3-packing is presented solely by a figure. Moreover, let ϕ_1 be an identity embedding ($\phi_1(x) = x$ for all $x \in V(G)$).

For the proof of our main result, we will need the following lemma which generalizes Lemma 6 from [9].

Lemma 1.11. *If a graph $G = C_{n_1} \cup C_{n_2} \cup \ldots \cup C_{n_k}$ has a 3-packing α such that the graph $\alpha_1(G) \oplus \alpha_2(G) \oplus \alpha_3(G)$ is not connected (a disconnected 3-packing), then G has another 3-packing α' such that the graph $\alpha'_1(G) \oplus \alpha'_2(G) \oplus \alpha'_3(G)$ is connected (a connected 3-packing). In particular, the graph G has two distinct 3-packings.*

Proof. Let α be the 3-packing with the smallest number of connected components. If $H = \alpha_1(G) \oplus \alpha_2(G) \oplus \alpha_3(G)$ is connected, we are done. Otherwise, let H_1, H_2 be two components of H.

Take a vertex x_1 of H_1 such that removing the two blue edges $x_1^- x_1$ and $x_1^+ x_1$ (where x_1^- and x_1^+ are neighbors of x_1 on the blue cycle in H_1) leaves H_1 connected. In a similar manner, take x_2, a vertex belonging to the component H_2. Note that such selections of x_1 and x_2 are always possible: it suffices to take, in H_1 and H_2, any vertex that is not a cut vertex (for example, the last vertex on the longest component path).

If now instead of the edges $x_1^- x_1$ and $x_1^+ x_1$ we add two blue edges $x_2^- x_1$ and $x_2^+ x_1$, and instead of the edges $x_2^- x_2$ and $x_2^+ x_2$ we add two blue edges $x_1^- x_2$ and $x_1^+ x_2$, we obtain a new 3-packing α' where two components H_1 and H_2 become one connected component. However, this is a contradiction with the choice of the 3-packing α. $\square$

2. CYCLES

In this section we prove the following lemma which will be useful in the remaining part of the proof:

Lemma 2.1. *Let C_n be a cycle of length n. For cycles C_3, C_4, C_5 and C_6, there is no 3-packing. The cycles C_7 and C_8 have unique 3-packing. For longer cycles, there exist at least two distinct 3-packings.*

Proof. Obviously, the cycles C_3, C_4, C_5 and C_6 do not have 3-packing because such a 3-packing is always a 6-regular graph. From Observation 1.9 we know that C_7 and C_8 have unique 3-packing.

We will denote by $C_n(a,b,c)$ the 6-regular circulant graph on n vertices with generators a, b and c, that is, the graph with vertex set $\mathbb{Z}_n = \{0,\dots,n-1\}$ and edge set $\{\{x,x+s\} : x \in \mathbb{Z}_n, s \in \{a,b,c\}\}$ (note that the addition is modulo n). In [7], Dean proved that every 6-regular circulant graph on n vertices with at least one generator of order n (with respect to the group $\mathbb{Z}_n$) has Hamiltonian cycle decomposition. Thus, it suffices to find, for any $n \geq 9$, two nonisomorphic 6-regular circulant graphs on n vertices with at least one generator of order n; this confirms that there exist two distinct 3-packings of a cycle with at least nine vertices.

To distinguish between 6-regular circulant graphs on $n \geq 9$ vertices with at least one generator of order n, we use their chromatic number. The following two results from [13] and [11] give information on the chromatic number of specific 6-regular circulant graphs:

Theorem 2.2. *Let $G = C_n(a,b,c)$ be a connected 6-regular circulant graph, where $n \geq 7$, $c = a+b$ or $n-c = a+b$ are pairwise distinct positive integers different from $n/2$. Let $\chi(G)$ be the chromatic number of G. Then*

(1) $\chi(G) = 7$ *if and only if* $G \cong K_7 \cong C_7(1,2,3)$,
(2) $\chi(G) = 6$ *if and only if* $G \cong C_{11}(1,2,3)$,
(3) $\chi(G) = 5$ *if and only if* $G \cong C_n(1,2,3)$ *and* $n \neq 7, 11$ *is not divisible by* 4, *or* G *is isomorphic to one of the following circulant graphs:* $C_{13}(1,3,4)$, $C_{17}(1,3,4)$, $C_{18}(1,3,4)$, $C_{19}(1,7,8)$, $C_{25}(1,3,4)$, $C_{26}(1,7,8)$, $C_{33}(1,6,7)$, $C_{37}(1,10,11)$,
(4) $\chi(G) = 3$ *if and only if* n *is divisible by* 3 *and none of* a, b, c *is divisible by* 3,
(5) $\chi(G) = 4$ *in all the remaining cases.*

Theorem 2.3. *Let G be a connected circulant graph of order n. Then G is bipartite if and only if n is even and all generators are odd.*

From Theorem 2.2 we have that there exist two nonisomorphic 6-regular circulant graphs of order $n \geq 9$ where $n \neq 11$ and $n \not\equiv 0 \pmod 4$: one can take, for example, $C_n(1,2,3)$ which has chromatic number equal to five and $C_n(1,4,5)$ which has chromatic number equal to three or four. The same theorem yields that $C_{11}(1,4,5)$ has chromatic number equal to four, and $C_{11}(1,2,3)$ has chromatic number equal to six. Hence, it remains to find two nonisomorphic 6-regular circulant of graphs of order $n \geq 9$ where n is divisible by four. Using Theorem 2.3, we know that $C_{4l}(1,3,5)$ is bipartite whereas $C_{4l}(1,3,4)$ is not bipartite. Therefore, there always exist two nonisomorphic 6-regular circulant graphs on $n \geq 9$ vertices with one generator of order n. □

3. GENERAL STRATEGY OF THE PROOF FOR 2-FACTORS WHERE $k \geq 2$

At first, we prove the following useful lemma about the existence of 3-packing of 2-factors which contain at least two cycles:

Lemma 3.1. *Let G be a vertex-disjoint union of $k \geq 2$ cycles. Then G has a 3-packing except when $G = 2C_3$.*

Proof. In the proof, we use the following Aigner and Brandt result from [2]:

Theorem 3.2. *Let H be a graph of order n with $\delta(H) \geq \frac{2n-1}{3}$. Then H contains any graph G of order at most n with $\Delta(G) = 2$ (as a subgraph).*

From Theorem 1.5 we know that the graph $2C_3$ is the only one which is not 2-packable, and therefore $2C_3$ also does not have a 3-packing; hence, we can assume that $n > 6$. From the fact that 2-packing of a 2-factor is always a 4-regular graph, we obtain that its complement H has $\delta(H) = n-1-4 = n-5$. From Theorem 3.2 we can see that H contains a packing of additional third copy of G if $\delta(H) \geq \frac{2n-1}{3}$. Therefore, for every $n \geq 14$ we have $n-5 = \delta(H) \geq \frac{2n-1}{3}$, and so contains a packing of additional third copy of G. This proves that every 2-factor G of order $n \geq 14$ has a 3-packing.

Using Observation 1.8 and the Oberwolfach problem solutions for all orders $n \leq 40$ (see [1] and [8]), we get that every 2-factor G of order $7 \leq n \leq 14$ except for $C_4 \cup C_5$, $2C_3 \cup C_5$ and $4C_3$ has a 3-packing. The 3-packings of $C_4 \cup C_5$, $2C_3 \cup C_5$ and $4C_3$ also exist, and they are presented in Section 5. □

Now, we present the general strategy of the remaining part of the proof of our main result. Assume that $G = C_{n_1} \cup C_{n_2} \cup \ldots \cup C_{n_k}$ is a vertex-disjoint union of $k \geq 2$ cycles, where $n = n_1 + \ldots + n_k$. Without loss of generality, assume that $n_1 \leq n_2 \leq \ldots \leq n_k$. Note that for all such 2-factors, except for $2C_3$, a 3-packing exists by Lemma 3.1. From Observation 1.9 and the Oberwolfach problem solutions for orders $n \in \{7, 8\}$ we know that 2-factors on seven and eight vertices have unique 3-packing. Therefore, we may assume that $n \geq 9$. We consider several cases according to k.

If $k = 2$ then $G = C_{n_1} \cup C_{n_2}$ and $n_1 \leq n_2$. If $n_1 \geq 7$, we have disconnected 3-packing of G which consists of two components. Each of these components we obtain as a 3-packing of a cycle from Lemma 2.1. Thus, by Lemma 1.11, the graph G has two distinct 3-packings. Therefore, we have to consider four families of 2-factors: $G = C_3 \cup C_x$ where $x \geq 6$, $G = C_4 \cup C_x$ where $x \geq 5$, $G = C_5 \cup C_x$ where $x \geq 5$ and $G = C_6 \cup C_x$ where $x \geq 6$. The uniqueness of 3-packing of 2-factors from these families is investigated in Section 4 for $x \geq 11$ and in Section 5 for $x \leq 10$.

If $k = 3$ then $G = C_{n_1} \cup C_{n_2} \cup C_{n_3}$ and $n_1 \leq n_2 \leq n_3$. We can divide G into two subgraphs $G_1 = C_{n_1} \cup C_{n_2}$ and $G_2 = C_{n_3}$. Thus, from Lemmas 2.1 and 3.1 we get a 3-packing of G_1 and G_2 except for the case $G = C_3 \cup C_3 \cup C_x$ where $x \geq 3$, and the following twelve 2-factors:

$$G = C_3 \cup C_4 \cup C_4, \quad G = C_3 \cup C_4 \cup C_5, \quad G = C_3 \cup C_4 \cup C_6,$$
$$G = C_4 \cup C_4 \cup C_4, \quad G = C_4 \cup C_4 \cup C_5, \quad G = C_4 \cup C_4 \cup C_6,$$
$$G = C_4 \cup C_5 \cup C_5, \quad G = C_4 \cup C_5 \cup C_6, \quad G = C_5 \cup C_5 \cup C_5,$$
$$G = C_5 \cup C_5 \cup C_6, \quad G = C_5 \cup C_6 \cup C_6, \quad G = C_6 \cup C_6 \cup C_6.$$

Therefore, by Lemma 1.11, the graph G has two distinct 3-packings. The uniqueness of 3-packing of 2-factors from the family $G = C_3 \cup C_3 \cup C_x$ where $x \geq 11$ is investigated in Section 4 and, in Section 5, we investigate 2-factors from the family $G = C_3 \cup C_3 \cup C_x$ where $x \in \{3, 4, \ldots, 11\}$ and the above mentioned twelve exceptional 2-factors.

If $k = 4$ then $G = C_{n_1} \cup C_{n_2} \cup C_{n_3} \cup C_{n_4}$ and $n_1 \leq n_2 \leq n_3 \leq n_4$. If at least two n_i (where $i \in \{1, 2, 3, 4\}$) are different from three then we can divide G into two parts $G = G_1 \cup G_2$ such that both G_1 and G_2 have 3-packing by Lemma 3.1. Therefore, by Lemma 1.11, the graph G has two distinct 3-packings. We argue similarly when $n_4 \geq 7$ (however, in this case, we need to use also Lemma 2.1). Thus, we are left with the following 2-factors: $G = C_3 \cup C_3 \cup C_3 \cup C_3$, $G = C_3 \cup C_3 \cup C_3 \cup C_4$, $G = C_3 \cup C_3 \cup C_3 \cup C_5$ and $G = C_3 \cup C_3 \cup C_3 \cup C_6$; the uniqueness of 3-packing of these four 2-factors is investigated in Section 5.

If $k = 5$ then $G = C_{n_1} \cup C_{n_2} \cup C_{n_3} \cup C_{n_4} \cup C_{n_5}$ and $n_1 \leq n_2 \leq n_3 \leq n_4 \leq n_5$. If $n_5 \geq 4$, we can divide G into two parts $G = G_1 \cup G_2$ such that $G_1 = C_{n_1} \cup C_{n_2} \cup C_{n_3}$ and $G_2 = C_{n_4} \cup C_{n_5}$ have 3-packing by Lemma 3.1. Therefore, by Lemma 1.11, we know that G has two distinct 3-packings. Thus, we have to investigate the uniqueness of 3-packing of $G = 5C_3$; this will be done in Section 5.

If $k \geq 6$ then $G = C_{n_1} \cup C_{n_2} \cup \ldots \cup C_{n_k}$ and $n_1 \leq n_2 \leq \ldots \leq n_k$. We can divide G into two parts $G = G_1 \cup G_2$ so that $G_1 = C_{n_1} \cup C_{n_2} \cup C_{n_3}$ and $G_2 = C_{n_4} \cup C_{n_5} \cup \ldots \cup C_{n_k}$. From Lemma 3.1 and the fact that $k \geq 6$, we have 3-packings of G_1 and G_2. Therefore, G has a disconnected 3-packing, and, from Lemma 1.11, we get the connected one.

4. FIVE PARTICULAR FAMILIES OF 2-FACTORS

In this section we present two distinct 3-packings of 2-factors from five families: $C_3 \cup C_x$, $C_4 \cup C_x$, $C_5 \cup C_x$, $C_6 \cup C_x$ and $C_3 \cup C_3 \cup C_x$ where $x \geq 11$. We use the construction approach. The first presented 3-packing of these 2-factors will contain a clique K_5 whereas the second one will not. At first, we present the 3-packing with a clique K_5.

We start our construction of 3-packing of 2-factors with a clique K_5 from the smallest graphs in each family, that is, the graphs $C_3 \cup C_{11}$, $C_4 \cup C_{11}$, $C_5 \cup C_{11}$, $C_6 \cup C_{11}$ and $C_3 \cup C_3 \cup C_{11}$. The 3-packing of these 2-factors is presented in Figures 1 and 2. For the clarity of drawings, some cycles connecting vertices marked with the same color and type of the marker are not drawn. More precisely, in Figure 1, in the 3-packing of $C_3 \cup C_{11}$, we have also the red cycle $v_1 v_4 v_{10} v_1$ and the blue cycle $v_2 v_{12} v_{11} v_2$. In the 3-packing of $C_4 \cup C_{11}$, we have also the red cycle $v_1 v_4 v_7 v_{10} v_1$ and the blue cycle $v_2 v_{12} v_{14} v_{11} v_2$. In the 3-packing of $C_5 \cup C_{11}$, we have also the red cycle $v_1 v_4 v_{13} v_7 v_{10} v_1$ and, in the 3-packing of $C_6 \cup C_{11}$, we have also the red cycle $v_1 v_4 v_{13} v_{16} v_7 v_{10} v_1$.

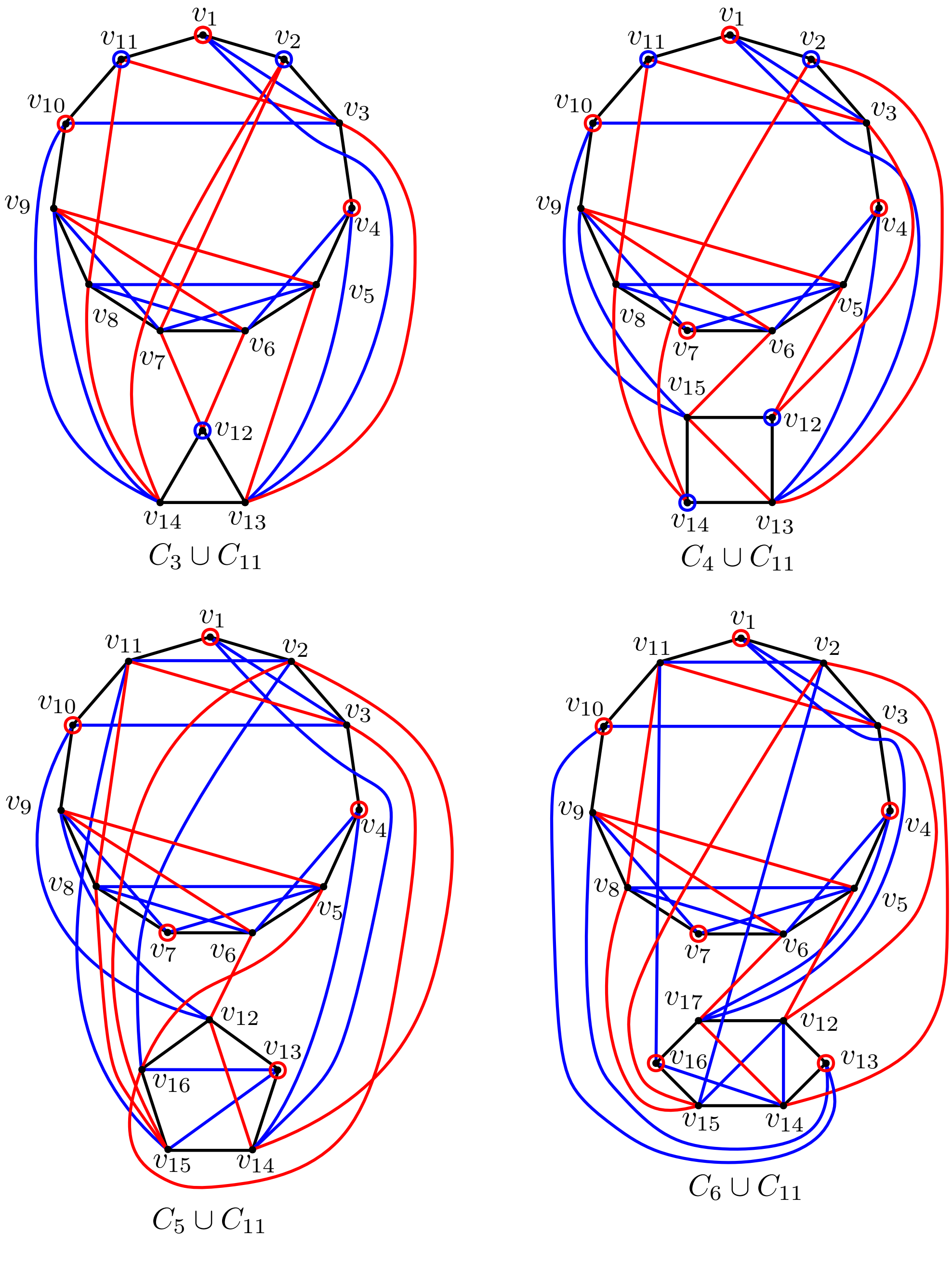

Fig. 1. The 3-packing of $C_3 \cup C_{11}$, $C_4 \cup C_{11}$, $C_5 \cup C_{11}$ and $C_6 \cup C_{11}$ with a clique K_5

Similarly, in Figure 2, in the 3-packing of $C_3 \cup C_3 \cup C_{11}$, we have also red cycles $v_1v_4v_{10}v_1$, $v_7v_{13}v_{15}v_7$ and blue cycles $v_2v_{14}v_{15}v_2$, $v_{11}v_{13}v_{16}v_{11}$. Note that each of these 3-packings contains a subgraph K_5 induced by vertices v_5, v_6, v_7, v_8 and v_9.

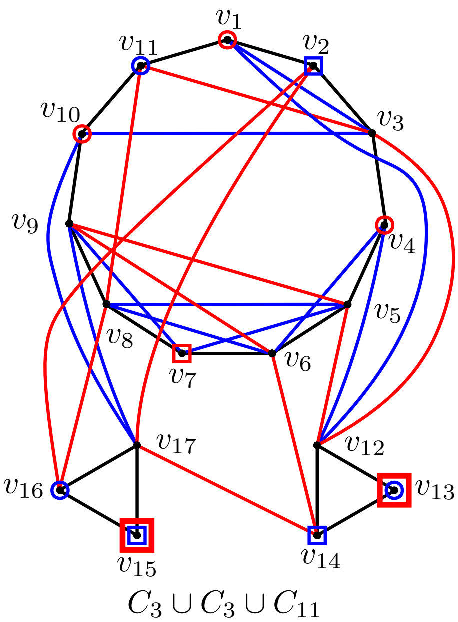

Fig. 2. The 3-packing of $C_3 \cup C_3 \cup C_{11}$ with a clique K_5

The presented 3-packings of the smallest graphs from these particular families are easily extendable to appropriate graphs from these families if the longest cycle C_x in the considered 2-factor has odd length. Note that in every already-presented 3-packing, the vertices v_{10}, v_{11}, v_1, v_2 and v_3 induce almost the same subgraph (up to the edge $v_{11}v_2$ which is not present in the 3-packing of $C_3 \cup C_3 \cup C_{11}$). Therefore, we introduce a common extension for these 3-packings in which we will change only the "upper part" of the 3-packing, which contains vertices v_{10}, v_{11}, v_1, v_2, v_3 and edges incident to them. The method of extension depends on the number of edges added to the longest cycle in each copy of the smallest 2-factor in the respective family. To increase the length of the longest cycle in each copy of the smallest 2-factor in these families by 2, 4 or 6, we use appropriate extension presented in Figure 3. For better understanding, we describe each of these extensions in detail.

Extension by 2. At first, we replace black edges v_2v_3 and $v_{11}v_{10}$ by black paths $v_2a_1v_3$ and $v_{11}b_1v_{10}$, respectively. Then, we replace red edge $v_{11}v_8$ by red edge b_1v_8. Next, we replace the red edge from the "bottom part" to v_3 by the red edge from the same vertex in the "bottom part" to a_1. Then, we replace the blue edge from the "bottom part" to v_1 by the blue edge from the same vertex in the "bottom part" to b_1. At the end, we add red edges b_1v_3, $v_{11}a_1$ and the blue path $v_1a_1b_1$.

Extension by 4. At first, we replace black edges v_2v_3 and $v_{11}v_{10}$ by black paths $v_2a_1a_2v_3$ and $v_{11}b_1b_2v_{10}$, respectively. Then, we replace the red edge from the "bottom part" to v_3 by the red edge from the same vertex in the "bottom part" to a_1. Next, we replace the blue edge from the "bottom part" to v_1 by the blue edge from the same vertex in the "bottom part" to b_2. Then, we remove the red edge $v_{11}v_3$. Next, we replace two red edges from the "bottom part" to v_2 by two red edges from the same vertices in the "bottom part" to a_2. At the end, we add the red path $a_1v_3b_1v_2b_2v_{11}$ and the blue path $v_1a_1b_1a_2b_2$.

Extension by 6. At first, we replace black edges v_2v_3 and $v_{11}v_{10}$ by black paths $v_2a_1a_2a_3v_3$ and $v_{11}b_1b_2b_3v_{10}$, respectively. Then, we replace the blue edge from the "bottom part" to v_1 by the blue edge from the same vertex in the "bottom part" to b_3. Then, we remove the red edge $v_{11}v_3$ and the blue edge v_1v_3. Next, we replace two red edges from the "bottom part" to v_2 by two red edges from the same vertices in the "bottom part" to a_3. At the end, we add the red path $v_3b_1v_2b_3a_2b_2a_1v_{11}$ and the blue path $v_3b_2a_3b_1a_2v_1a_1b_3$.

Now, we describe how to create the 3-packing of a 2-factor with the longest cycle C_{11+2t}, for $t > 3$, from the smallest 2-factor in the respective family. At first, we replace black edges v_2v_3 and $v_{11}v_{10}$ by black paths $v_2a_1a_2\dots a_tv_3$ and $v_{11}b_1b_2\dots b_tv_{10}$, respectively. Then, we replace the blue edge from the "bottom part" to v_1 by the blue edge from the same vertex in the "bottom part" to b_t. Then, we remove the red edge $v_{11}v_3$ and the blue edge v_1v_3. At the end, we add the red path $v_3b_1a_tb_ta_{t-1}b_{t-1}\dots a_2b_2a_1v_{11}$ and the blue path $v_3b_{t-1}a_tb_{t-2}a_{t-1}\dots b_2a_3b_1a_2v_1a_1b_t$. This extension of 3-packing is also presented in Figure 3.

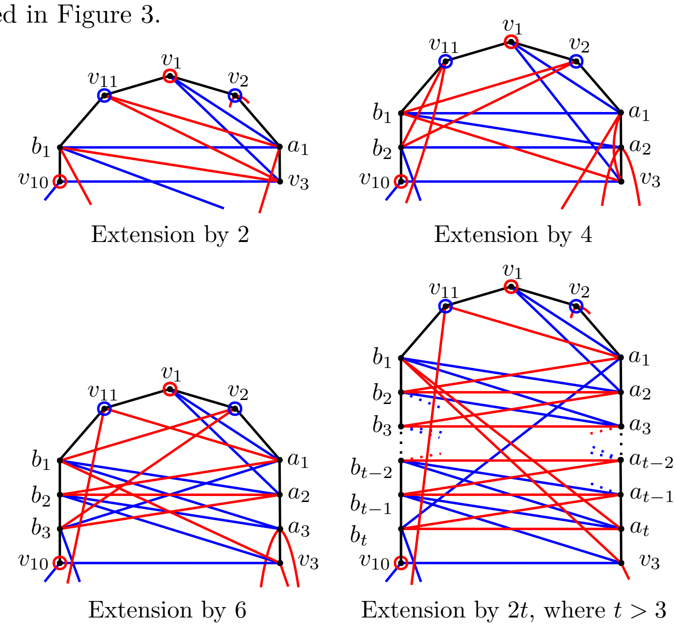

Fig. 3. The extensions of 3-packings of 2-factors from five particular families when the longest cycle has odd length

Now, we show how to obtain a 3-packing of a 2-factor from each of these families, if the longest cycle in 2-factor has an even length x. We take the 3-packing of a 2-factor in which the longest cycle has length $x-1$. We replace: the black edge v_1v_2 by the black path $v_1v_cv_2$, the blue edge $v_{10}v_3$ by the blue path $v_{10}v_cv_3$, the red edge $v_{11}v_8$ by the red path $v_{11}v_cv_8$ (if $x = 14$, we replace the red edge $v_{11}a_1$ by the red path $v_{11}v_ca_1$). Thus, we obtain a 3-packing of a 2-factor from each of these families if the longest cycle in 2-factor has an even length. Note that in each 3-packing which we obtain using the above extensions we have induced subgraph K_5 on vertices v_5, v_6, v_7, v_8 and v_9.

Next, for 2-factors from these five particular families, we present K_5-free 3-packings, i.e., 3-packings which do not contain a clique K_5. Note that $m \geq 2$ edges of a graph are independent if no two of them share a common vertex. The following observation is particularly useful for the construction:

Observation 4.1. *Let B be a union of cycles, let $q \geq 3$, and let G be an instance of a K_5-free 3-packing of $B \cup C_q$. If there are three independent edges e_1, e_2, and e_3 on C_q in the black, the blue, and the red copy of $B \cup C_q$, then there is a K_5-free 3-packing of $B \cup C_{q+1}$.*

Proof. Let $e_1 = x_1x_2$, $e_2 = y_1y_2$, and $e_3 = z_1z_2$. Let w be a new vertex. Replace edges x_1x_2, y_1y_2, z_1z_2 with edges $x_1w, wx_2, y_1w, wy_2, z_1w, wz_2$. The new graph G' is a 3-packing of $C_3 \cup C_{q+1}$. Moreover, since no edges were added between the vertices of $V(G)$, the 5-clique, if there is one, contains w. Let there be a 5-clique A containing w in G'. This clique contains four neighbors of w, and, therefore, it contains two neighbors of w which were adjacent in G but are not adjacent in G'; this, however, contradicts the fact that A is a clique. □

Note that, instead of adding one new vertex, it is possible to add k vertices at once, if there are k pairwise edge-disjoint matchings, each consisting of three edges of G lying on C_q in the black, the blue, and the red subgraph of G, respectively. We will use this fact later.

Consider first the 3-packing of $C_3 \cup C_{11}$ in Figure 4; note that this graph is K_5-free, which can be checked using computer. We now show how to extend the 3-packing in Figure 4 to a K_5-free 3-packing of $C_3 \cup C_{11+4t}$ for some positive integer t. Denote by G the graph in Figure 4. Let G' be a graph obtained from G after removing edges $v_3v_4, v_3v_{12}, v_4v_5, v_4v_{10}, v_5v_9, v_8v_9, v_8v_{13}, v_9v_{10}$ (dashed edges in Figure 4), adding new vertices a_i, b_i, c_i, d_i for $i \in \{1, \ldots, t\}$, and adding edges of the (black) paths $v_3(a_i)_{i=1}^t v_9$, $v_5(b_i)_{i=1}^t v_4$, $v_{10}(c_i)_{i=1}^t v_9$ and $v_8(d_i)_{i=1}^t v_9$, (blue) paths $v_{10}(a_ic_i)_{i=1}^t v_4$ and $v_5(d_ib_i)_{i=1}^t v_9$, and (red) paths $v_3(b_ia_i)_{i=1}^t v_{12}$ and $v_8(c_id_i)_{i=1}^t v_{13}$. For an overview of the added part, see Figure 5.

In the following, we refer to vertices from $\{v_1, \ldots, v_{14}\}$ as old vertices, and other vertices of G' as new vertices.

It follows from the construction that G' is a 3-packing of $C_3 \cup C_{11+4t}$, however, the absence of K_5 in G' is not clear. Therefore, suppose to the contrary that there is a copy of K_5 in G'. Let H be a subgraph of G' induced on new vertices and their neighbors (see Figure 5). Note that no edge between two old vertices was added in the construction;

hence, if there is a clique on five vertices, it contains at least one new vertex. Thus, each 5-clique in G' is a 5-clique in H. Note also that $H - a_1$ is 3-degenerate (consider for example the ordering $v_{12}, v_{13}, v_3, v_{10}, v_8, v_5, c_1, d_1, \ldots, c_t, d_t, v_9, b_1, a_2, b_2, \ldots a_t, b_t, v_4$); hence, if there is a 5-clique in H, then it contains a_1. However, a_1 and its neighbors induce a planar graph, see Figure 6. Hence, there is no 5-clique in G.

Using Observation 4.1 for all three (pairwise edge-disjoint) matchings $v_1v_3, v_6v_{11}, v_7v_8$, $v_1v_5, v_6v_{13}, v_{10}v_{11}$, and $v_1v_{11}, v_3v_{10}, v_9v_{12}$ at once, we get that there is a K_5-free 3-packing of $C_3 \cup C_{11+4t+q}$ for every nonnegative integer t and every $q \in \{0, 1, 2, 3\}$. This covers all the cases of 3-packings of $C_3 \cup C_x$, $x \geq 11$. Moreover, such a construction does not remove any of the edges of the initial black, blue, or red copy of C_3; this will be useful to extend created K_5-free 3-packings of $C_3 \cup C_x$ to K_5-free 3-packings of $C_y \cup C_x$ for $y \in \{4, 5, 6\}$. To obtain a 3-packing of $C_4 \cup C_x$, apply Observation 4.1 for $B = C_x$, $q = 4$, $e_1 = v_{12}v_{13}$, $e_2 = v_2v_{11}$, and $e_3 = v_4v_{14}$; for later use, denote the newly added vertex by w_1.

Note that there are no larger sets of matchings of desired properties in the 3-packing of $C_3 \cup C_x$ that could be used to extend them to a 3-packing of $C_5 \cup C_x$ or $C_6 \cup C_x$ at once. However, we can do it in steps. Observe that, in the created 3-packing of $C_4 \cup C_x$, the black edge w_1v_{13}, the blue edge $v_{11}v_{12}$, and the red edge v_2v_{14} form a matching, and we can make use of Observation 4.1 to obtain a K_5-free 3-packing of $C_5 \cup C_x$; denote the newly added vertex by w_2 (see Figure 7). To obtain a 3-packing of $C_6 \cup C_x$, simply repeat the previous step for edges $v_{13}v_{14}, v_{11}w_2, v_2v_4$.

Finally, we show how to obtain a K_5-free 3-packing of $C_3 \cup C_3 \cup C_x$ for $x \geq 11$. The process is similar to the case of K_5-free 3-packings of $C_3 \cup C_x$. We start with the initial 3-packing G of $C_3 \cup C_3 \cup C_{11}$ displayed in Figure 8. Using a computer check, one can find that such a 3-packing is K_5-free (in this case, the size of the maximum clique is three).

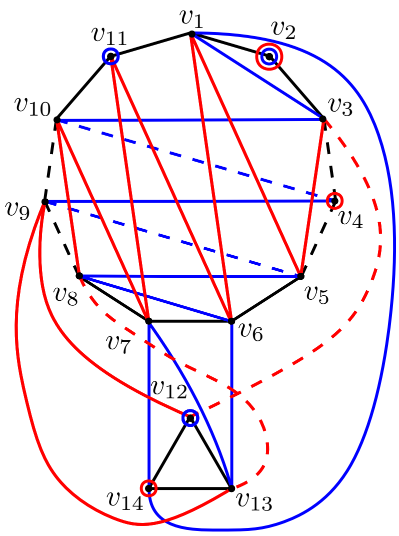

Fig. 4. K_5-free 3-packing of $C_3 \cup C_{11}$

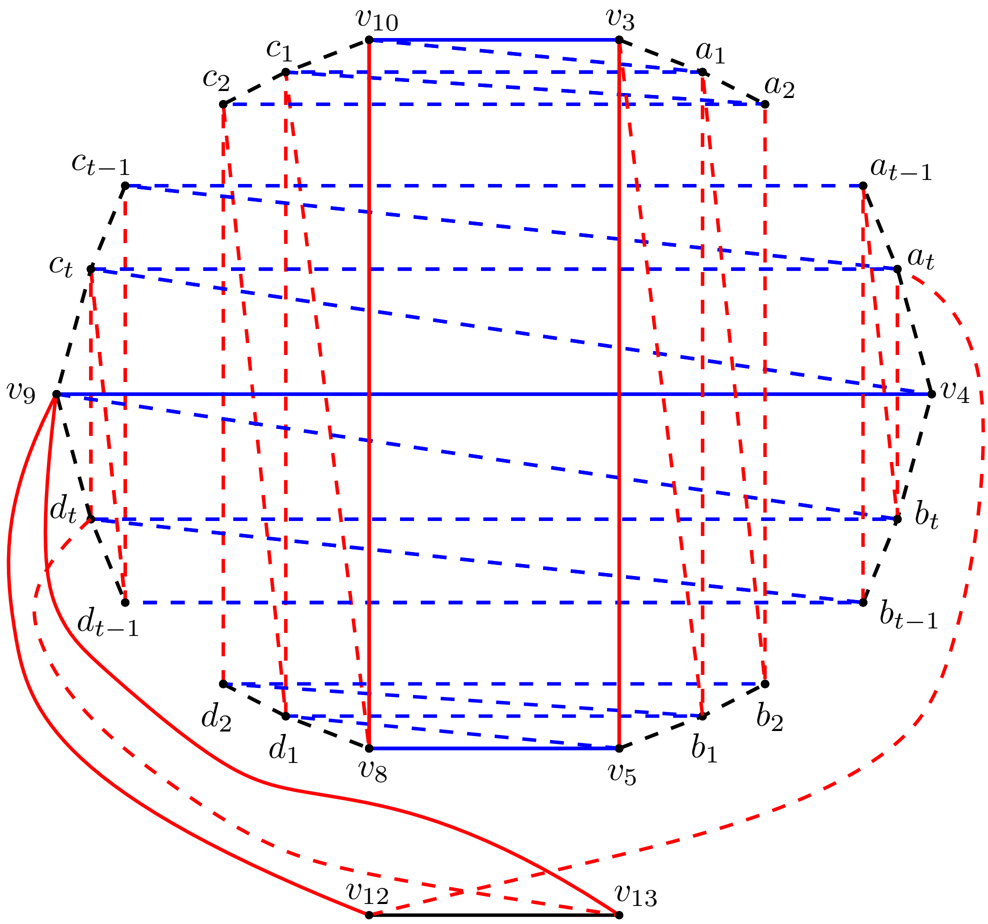

Fig. 5. The subgraph H of the K_5-free 3-packing of $C_3 \cup C_{11+4t}$ induced on $4t$ added vertices and their neighbors. Dashed edges are newly added edges, full edges are old edges

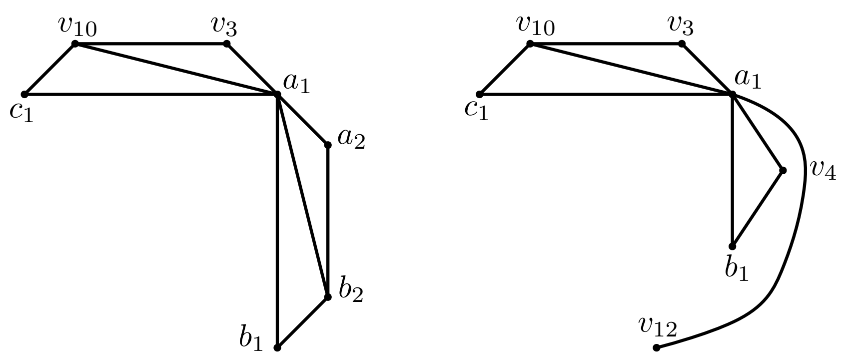

Fig. 6. The subgraph induced on $N(a_1) \cup \{a_1\}$ for $t \geq 2$ (left) and $t = 1$ (right)

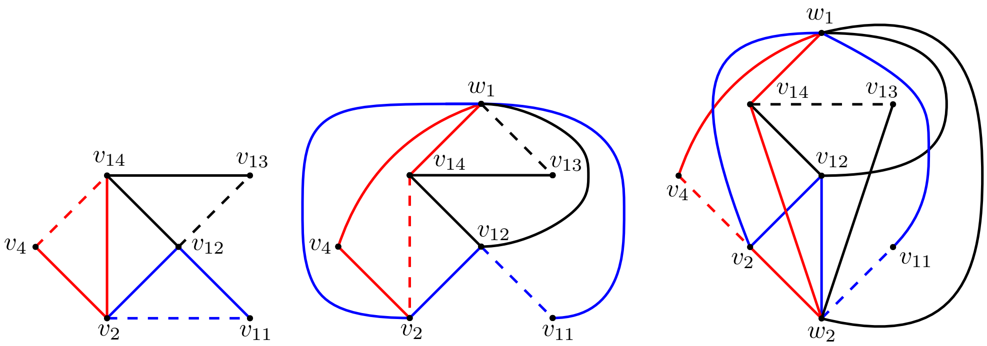

Fig. 7. Extensions of monochromatic triangles. Dashed edges are edges of a matching for which Observation 4.1 is used

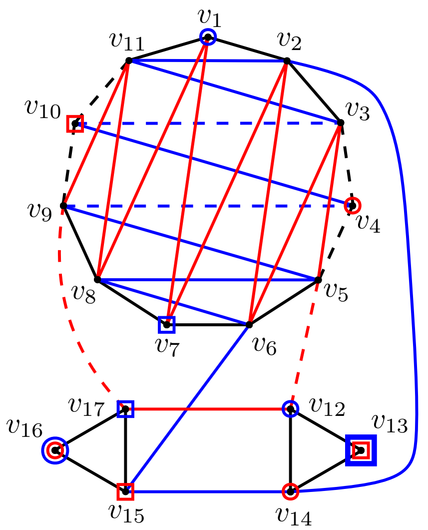

Fig. 8. K_5-free 3-packing of $C_3 \cup C_3 \cup C_{11}$

We show how to extend G to a K_5-free 3-packing of $C_3 \cup C_3 \cup C_{11+4t}$ for some positive integer t, using a similar approach as previously. Let G' be a graph obtained from G by adding $4t$ new vertices, namely a_i, b_i, c_i, d_i for $i \in \{1, \dots, t\}$, removing the edges $v_3v_4, v_3v_{10}, v_4v_5, v_4v_9, v_5v_{12}, v_9v_{10}, v_9v_{17}, v_{10}v_{11}$ (dashed edges in Figure 8), and adding (black) paths $v_{11}(a_i)_{i=1}^t v_{10}$, $v_3(b_i)_{i=1}^t v_4$, $v_9(c_i)_{i=1}^t v_{10}$, and $v_5(d_i)_{i=1}^t v_4$, (blue) paths $v_{11}(b_ia_i)_{i=1}^t v_4$ and $v_5(c_id_i)_{i=1}^t v_{10}$, and (red) paths $v_9(a_ic_i)_{i=1}^t v_{17}$ and $v_5(b_id_i)_{i=1}^t v_{12}$ (see Figure 9).

Clearly, the presented construction creates a 3-packing of $C_3 \cup C_3 \cup C_{11+4t}$. Suppose that a 5-clique A was created in the process. Since no edge between two old vertices (that is, the vertices from $\{v_1, \dots, v_{17}\}$) was added, at least one of the vertices of A is a new vertex. Hence, the 5-clique is present in a subgraph H induced on new vertices and their neighbors, see Figure 9.

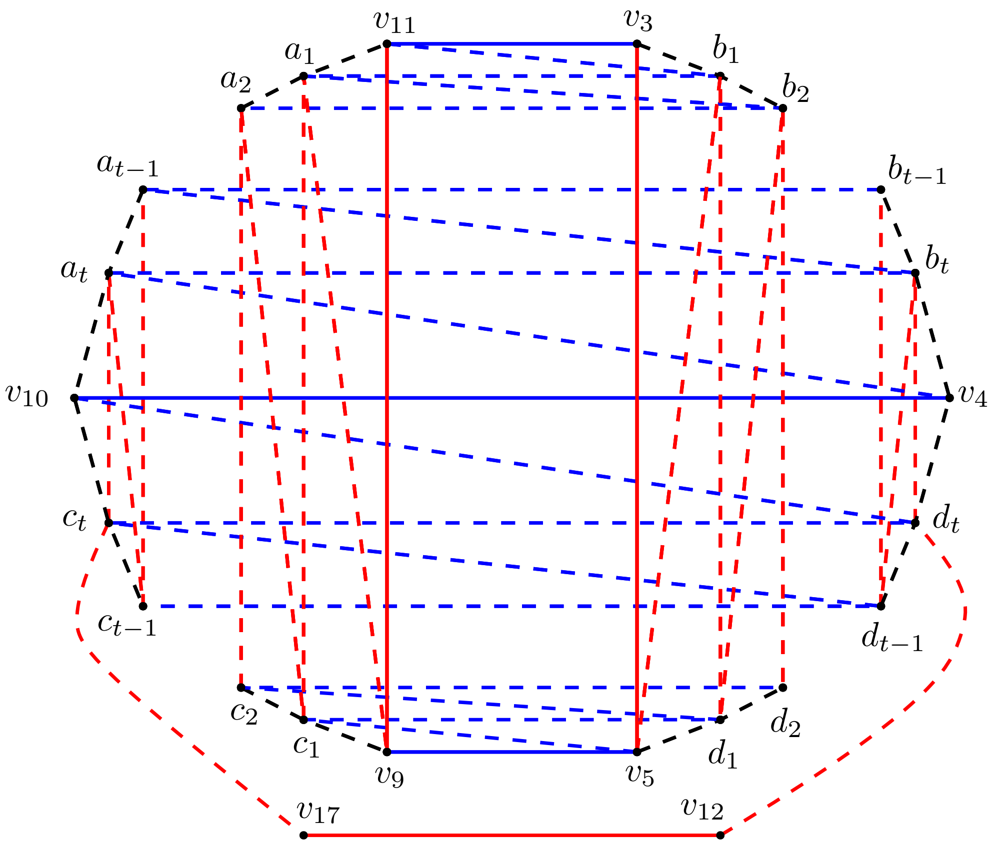

Fig. 9. The subgraph H induced on new vertices and their neighborhood in the K_5-free 3-packing of $C_3 \cup C_3 \cup C_{11+4t}$. Dashed edges are newly added edges

Vertices v_{12} and v_{17} are of degree two in H, and the vertex v_3 is of degree three in H, hence, none of them is in A.

Vertices v_4, v_9, v_{10} and v_{11} are of degree four in H. Thus, if any of them is contained in the 5-clique A, then all its neighbors are from A; however, for each of these vertices, there are two of its neighbors that are not adjacent. Namely, for v_9, the vertices v_5 and v_{11} are not adjacent, for v_{11}, the vertices v_3 and v_9 are not adjacent, and, for v_4 and v_{10}, the vertices a_t and d_t are not adjacent. Since the degree of v_5 in H is five and at least two of its neighbors are not from the 5-clique A (namely v_3 and v_9), we get that $v_5 \notin A$. Hence, A contains only new vertices. However, the subgraph of G' induced on new vertices is 3-degenerate (consider the vertex ordering $b_1, a_1, b_2 \ldots, a_t, c_1, d_1, c_2, \ldots, d_t$) and, therefore, it does not contain a 5-clique.

Now, we show how to extend the constructed K_5-free 3-packing of $C_3 \cup C_3 \cup C_{11+4t}$ to a K_5-free 3-packing of $C_3 \cup C_3 \cup C_{11+4t+q}$ for $q \in \{1, 2, 3\}$. In all cases, use Observation 4.1 for three (pairwise edge-disjoint) matchings $v_1v_2, v_{14}v_{15}, v_3v_5$, $v_2v_3, v_6v_{15}, v_9v_{11}$, and $v_6v_7, v_5v_8, v_9v_{17}$. This completes the proof in case of five particular families of 2-factors.

5. REMAINING SMALL CASES

This section contains the discussion on the uniqueness of 3-packings of small 2-factors which were not treated by general constructions in Section 4. First, note that the 3-packing of $3C_3$ is unique, as it corresponds to Steiner triple system STS(9) on nine points (equivalently, to the affine plane of order three), which is unique. For the remaining 46 small 2-factors, Table 1 (Appendix) contains description of two distinct 3-packings of particular 2-factors; each of three 2-factors is presented as a collection of sequences of vertices of its cycles. Moreover, we present two distinct 3-packings of these 2-factors for which the Oberwolfach problem has no solution in Figures 10, 11 and 12.

For computer-assisted finding of these distinct 3-packings, we used Wolfram Mathematica computer algebra system [17] with its graph theory procedures. About the half of cases was solved using the following strategy: considering a 2-factor $H \cong C_{n_1} \cup \ldots \cup C_{n_k}$, we first removed from the complete graph $K_{n_1+\ldots+n_k}$ the cycles $(1, \ldots, n_1), (n_1+1, n_1+2, \ldots, n_1+n_2), \ldots, (n_1+\ldots+n_{k-1}+1, \ldots, n_1+n_2+\ldots+n_k)$. In the resulting graph G_1, we were looking for several distinct subgraphs isomorphic to H (using the procedure `FindIsomorphicSubgraph[` G `,` H `,` p `]` which allows to find either all, or at most p distinct copies of H in G). Among the graphs resulted from removing these subgraphs from G_1, we were looking for two nonisomorphic graphs G_2', G_2'', and, in them, we again looked for distinct subgraphs isomorphic to H. Finally, in two collections of graphs obtained from G_2', G_2'' by third removal of isomorphic copies of H, we were able to find two nonisomorphic graphs; their complements yielded the desired distinct 3-packings.

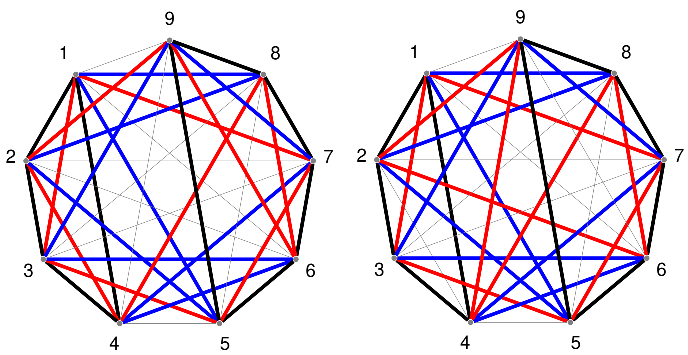

Fig. 10. Two distinct 3-packings of $C_4 \cup C_5$

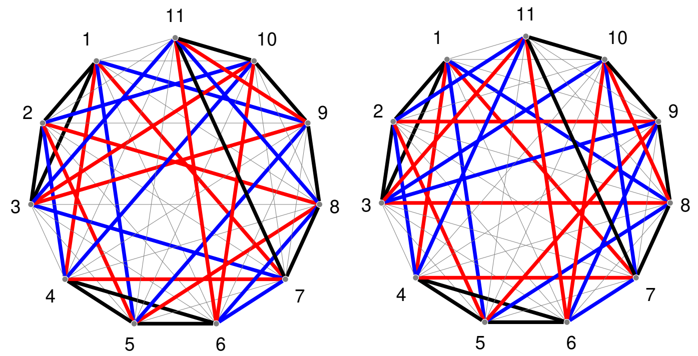

Fig. 11. Two distinct 3-packings of $2C_3 \cup C_5$

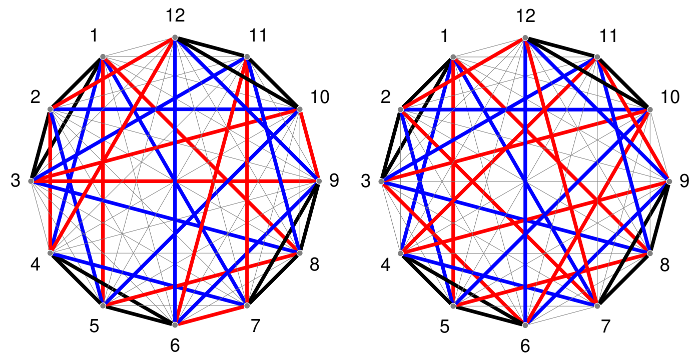

Fig. 12. Two distinct 3-packings of $4C_3$

We have to notice that this strategy failed when $k = 3, n_1 \geq 4, n_2 \geq 5$ and, also, when $k \geq 4$: the procedure `FindIsomorphicSubgraph[` G_1 `,` H `,` p `]` was able to find only at most three distinct copies of H in G_1 (higher values of p resulted in computation crash), and further attempts to look for copies of H in G_2', G_2'' have led to computation crash or to isomorphic graphs. To overcome these obstacles, we have generated, for each of the remaining cases, a collection of distinct 2-packings of H using the code in Python (which took a fixed 2-factor, then renamed its vertices using a random permutation to obtain another 2-factor with the same cycle structure, and then checked whether these two 2-factors are edge disjoint and forming together a 4-regular graph). These 2-packings of H were first removed from $K_{n_1+\ldots+n_k}$ and, in each of the obtained graphs, a single third copy of H was searched for (again using `FindIsomorphicSubgraph[]` procedure). Among the graphs resulted from this removal, we searched for a pair of nonisomorphic ones (again, their complements yielded the desired 3-packings). More details and the corresponding code can be found in Appendix.

Acknowledgements

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-23-0191 (Madaras, Onderko) and by the Science Grant Agency – project VEGA 1/0574/21 (Madaras, Onderko).

REFERENCES

[1] P. Adams, D. Bryant, *Two-factorisations of complete graphs of orders fifteen and seventeen*, Australas. J. Comb. **35** (2006), 113–118.

[2] M. Aigner, S. Brandt, *Embedding arbitrary graphs of maximum degree two*, J. London Math. Soc. **(2)** (1993), 39–51.

[3] B. Bollobás, S.E. Eldridge, *Packings of graphs and applications to computational complexity*, J. Combin. Theory Ser. B **25** (1978), 105–124.

[4] D. Burns, S. Schuster, *Every* $(p, p-2)$ *graph is contained in its complement*, J. Graph Theory **1** (1977), 277–279.

[5] D. Burns, S. Schuster, *Embedding* $(n, n-1)$ *graphs in their complements*, Israel J. Math. **30** (1978), 313–320.

[6] C.J. Colbourn, J.H. Dinitz (eds), *The CRC Handbook of Combinatorial Designs*, 2nd ed., CRC Press, Boca Raton, 2006.

[7] M. Dean, *On Hamilton cycle decomposition of* 6*-regular circulant graphs*, Graphs Combin. **22** (2006), 331–340.

[8] A. Deza, F. Franek, W. Hua, M. Meszka, A. Rosa, *Solutions to the Oberwolfach problem for orders 18 to 40*, J. Comb. Math. Comb. Comput. **74** (2010), 95–102.

[9] I. Grzelec, M. Pilśniak, M. Woźniak, *A note on uniquely embeddable* 2*-factors*, Appl. Math. Comput. **468** (2024), 128505.

[10] A. Hagberg, P. Swart, D. Schult, *Exploring network structure, dynamics, and function using NetworkX* (No. LA-UR-08–05495; LA-UR-08–5495). Los Alamos National Lab.(LANL), 2008, Los Alamos, NM (United States).

[11] C. Heuberger, *On planarity and colorability of circulant graphs*, Discrete Math. **268** (2003), 153–169.

[12] A.J.W. Hilton, M. Johnson, *Some results on the Oberwolfach problem*, J. London Math. Soc. **64** (2001), 513–522.

[13] M. Meszka, R. Nedela, A. Rosa, *Circulants and the chromatic index of Steiner triple systems*, Math. Slovaca **56** (2006) 371–378.

[14] J. Otfinowska, M. Woźniak, *A Note on Uniquely Embeddable Forests*, Discuss. Math. Graph Th. **33** (2013), no. 1, 193–201.

[15] N. Sauer, J. Spencer, *Edge disjoint placement of graphs*, J. Combin. Theory Ser. B **25** (1978), 295–302.

[16] H. Wang, N. Sauer, *Packing three copies of a tree into a complete graph*, European J. Combin. **14** (1993), 137–142.

[17] Wolfram Research, Inc., *Mathematica, Version* 14.1, Champaign, IL (2024).

[18] M. Woźniak, *Packing of graphs*, Diss. Math. **362** (1997), pp. 78.

[19] M. Woźniak, *A note on uniquely embeddable graphs*, Discuss. Math. Graph Theory **18** (1998), 15–21.

[20] M. Woźniak, *Packing of graphs and permutation – a survey*, Discrete Math. **276** (2004), 379–391.

[21] M. Woźniak, A.P. Wojda, *Triple placement of graphs*, Graphs Combin. **9** (1993), 85–91.

[22] H.P. Yap, *Packing of graphs – a survey*, Discrete Math. **72** (1988), 395–404.

APPENDIX

In this section, we present parts of the code and results of finding distinct packings for the finite number of small 2-factors considered in Section 5. As mentioned in that section, we mostly used Wolfram Mathematica built-in functions `FindIsomorphicSubgraph[` G_1 `,` H `,` p `]`. The code for a particular case of two distinct 3-packing of $C_4 \cup C_4 \cup C_5$ is listed here:

```
G = Graph[CompleteGraph[13], VertexLabels -> "Name"];

G1 = EdgeDelete[G, {
    1 \[UndirectedEdge] 2, 2 \[UndirectedEdge] 3,
    3 \[UndirectedEdge] 4, 4 \[UndirectedEdge] 1,
    5 \[UndirectedEdge] 6, 6 \[UndirectedEdge] 7,
    7 \[UndirectedEdge] 8, 8 \[UndirectedEdge] 5,
    9 \[UndirectedEdge] 10, 10 \[UndirectedEdge] 11,
    11 \[UndirectedEdge] 12, 12 \[UndirectedEdge] 13,
    13 \[UndirectedEdge] 9
    }];

list445inG1 = FindIsomorphicSubgraph[G1,
    GraphDisjointUnion[CycleGraph[4],CycleGraph[4],CycleGraph[5]]
    ];

G2 = EdgeDelete[G1, EdgeList[list445inG1[[1]]]];

list445inG2 = FindIsomorphicSubgraph[G2,
    GraphDisjointUnion[CycleGraph[4],CycleGraph[4],CycleGraph[5]],
    50];

t = False;
For[i = 1, i <= 50 && Not[t], i++,
    For[j = i + 1, j <= 50 && Not[t], j++,
        If[Not[IsomorphicGraphQ[listG2[[i]], listG2[[j]]]],
            Print[i, "␣", j, "␣", listG2[[i]], "␣", listG2[[j]]];
            t = True]
    ]
]
```

For cases different from $C_4 \cup C_4 \cup C_5$, the definition of `G1` and the length of each `CycleGraph` in the definition of `list445inG2` need to be adjusted.

We used the above code to find 50 different subgraphs of `G2` isomorphic to the given 2-factor, as seen in the provided code on the line when `list445inG2` is initialized. However, this approach mostly failed in some considered cases (probably due to sizes of considered graphs) and the function resulted only in the list of 3 subgraphs, which was insufficient for our search.

To overcome this obstacle, we constructed an easy Python code to generate (usually 50, but in some cases fewer) nonisomorphic 2-packings of given 2-factors. The code for such a generation is provided below. Note that `G1` is the original 2-factor given as an instance of Graph from NetworkX package [10] (with the vertices denoted by $0, \ldots, n-1$). This method returns (in most cases) the list of 50 (random) nonisomorphic complements of 2-packings (represented as strings in Graph6 graph format); this list is then exported to a .g6 file which can be imported as a list of graphs in Wolfram Mathematica. After that, the third copy of the given 2-factor was found in these graphs using `FindIsomorphicSubgraph[]` in Wolfram Mathematica, as well as the checking the resulting 3-packings for their nonisomorphism.

```
import networkx as nx
import numpy as np
import time


def two_packing_generator(G1):
    n = nx.number_of_nodes(G1)

    generated = []
    generated_g6 = []

    timeout = time.time() + 60
    while len(generated) < 50 and time.time() < timeout:
        G2 = G1.copy()
        p = dict(zip(range(n), list(np.random.permutation(n))))
        G2 = nx.relabel_nodes(G2, p)

        if any([G2.has_edge(*e) for e in G1.edges()]):
            # graphs are not edge-disjoint
            continue

        R = nx.complete_graph(n)
        R.remove_edges_from(list(G2.edges()) + list(G1.edges()))

        if any([nx.is_isomorphic(R, G) for G in generated]):
            # R is isomorphic to something we already have
            continue

        generated += [R]
        generated_g6 += [str(nx.to_graph6_bytes(R))
                         .removeprefix("b'>>graph6<<")
                         .removesuffix("\\n\'")
                         .replace("\\\\", "\\")]

    return generated_g6
```

The result of this code is a table which provides, for each considered 2-factor, two its distinct 3-packings (the individual cycles of these three 2-factors are represented by sequences of vertices).

Table 1
Two distinct 3-packings of each of 46 small 2-factors

2-factor	First 3-packing	Second 3-packing
$C_3 \cup C_6$	(1, 2, 3), (4, 5, 6, 7, 8, 9) (3, 4, 6), (1, 8, 2, 5, 7, 9) (2, 6, 9), (1, 4, 8, 5, 3, 7)	(1, 2, 3), (4, 5, 6, 7, 8, 9) (3, 4, 6), (1, 5, 9, 7, 2, 8) (2, 6, 9), (1, 4, 8, 5, 3, 7)
$C_3 \cup C_7$	(1, 2, 3), (4, 5, 6, 7, 8, 9, 10) (1, 4, 6), (2, 5, 3, 8, 10, 7, 9) (1, 5, 7), (2, 4, 8, 6, 9, 3, 10)	(1, 2, 3), (4, 5, 6, 7, 8, 9, 10) (1, 4, 6), (2, 5, 7, 9, 3, 8, 10) (1, 5, 8), (2, 4, 3, 7, 10, 6, 9)
$C_3 \cup C_8$	(1, 2, 3), (4, 5, 6, 7, 8, 9, 10, 11) (1, 4, 6), (2, 5, 3, 7, 9, 11, 8, 10) (1, 5, 7), (2, 8, 3, 9, 4, 10, 6, 11)	(1, 2, 3), (4, 5, 6, 7, 8, 9, 10, 11) (1, 4, 6), (2, 5, 3, 8, 10, 7, 11, 9) (1, 5, 7), (2, 4, 8, 11, 3, 9, 6, 10)
$C_3 \cup C_9$	(1, 2, 3), (4, 5, 6, 7, 8, 9, 10, 11, 12) (1, 4, 6), (2, 5, 3, 7, 9, 11, 8, 10, 12) (1, 5, 7), (2, 4, 8, 12, 9, 3, 10, 6, 11)	(1, 2, 3), (4, 5, 6, 7, 8, 9, 10, 11, 12) (1, 4, 6), (2, 5, 3, 7, 9, 11, 8, 12, 10) (1, 5, 7), (2, 4, 8, 10, 3, 9, 12, 6, 11)
$C_3 \cup C_{10}$	(1, 2, 3), (4, 5, 6, 7, 8, 9, 10, 11, 12, 13) (1, 4, 6), (2, 5, 3, 7, 9, 11, 8, 12, 10, 13) (1, 5, 7), (2, 4, 3, 8, 10, 6, 11, 13, 9, 12)	(1, 2, 3), (4, 5, 6, 7, 8, 9, 10, 11, 12, 13) (1, 4, 6), (2, 5, 3, 7, 9, 11, 8, 13, 10, 12) (1, 5, 7), (2, 4, 3, 6, 10, 8, 12, 9, 13, 11)
$C_4 \cup C_5$	(1, 2, 3, 4), (5, 6, 7, 8, 9) (1, 3, 5, 7), (2, 4, 8, 6, 9) (1, 5, 2, 8), (3, 6, 4, 7, 9)	(1, 2, 3, 4), (5, 6, 7, 8, 9) (1, 3, 5, 7), (2, 6, 8, 4, 9) (1, 5, 2, 8), (3, 6, 4, 7, 9)
$C_4 \cup C_6$	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10) (1, 3, 5, 7), (2, 4, 8, 10, 6, 9) (1, 5, 2, 8), (3, 6, 4, 9, 7, 10)	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10) (1, 3, 5, 7), (2, 6, 8, 10, 4, 9) (1, 5, 2, 8), (3, 9, 6, 4, 7, 10)
$C_4 \cup C_7$	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10, 11) (1, 3, 5, 7), (2, 4, 6, 9, 11, 8, 10) (1, 5, 2, 8), (3, 6, 10, 4, 9, 7, 11)	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10, 11) (1, 3, 5, 7), (2, 4, 8, 10, 6, 9, 11) (1, 5, 2, 6), (3, 8, 11, 4, 9, 7, 10)
$C_4 \cup C_8$	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10, 11, 12) (1, 3, 5, 7), (2, 4, 6, 8, 10, 12, 9, 11) (1, 5, 2, 6), (3, 9, 4, 10, 7, 11, 8, 12)	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10, 11, 12) (1, 3, 5, 7), (2, 4, 6, 9, 11, 8, 10, 12) (1, 5, 2, 6), (3, 8, 12, 9, 4, 10, 7, 11)
$C_4 \cup C_9$	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10, 11, 12, 13) (1, 3, 5, 7), (2, 4, 6, 8, 10, 12, 9, 11, 13) (1, 5, 2, 6), (3, 7, 9, 13, 10, 4, 11, 8, 12)	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10, 11, 12, 13) (1, 3, 5, 7), (2, 4, 6, 8, 10, 12, 9, 13, 11) (1, 5, 2, 6), (3, 7, 9, 11, 4, 10, 13, 8, 12)
$C_4 \cup C_{10}$	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10, 11, 12, 13, 14) (1, 3, 5, 7), (2, 4, 6, 8, 10, 12, 9, 13, 11, 14) (1, 5, 2, 6), (3, 7, 4, 9, 11, 8, 12, 14, 10, 13)	(1, 2, 3, 4), (5, 6, 7, 8, 9, 10, 11, 12, 13, 14) (1, 3, 5, 7), (2, 4, 6, 8, 10, 12, 9, 14, 11, 13) (1, 5, 2, 6), (3, 7, 4, 8, 11, 9, 13, 10, 14, 12)
$C_5 \cup C_5$	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10) (1, 3, 5, 6, 8), (2, 4, 9, 7, 10) (1, 4, 6, 2, 7), (3, 8, 10, 5, 9)	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10) (1, 3, 5, 2, 6), (4, 8, 10, 7, 9) (1, 4, 7, 2, 10), (3, 6, 8, 5, 9)
$C_5 \cup C_6$	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11) (1, 3, 5, 2, 4), (6, 8, 10, 7, 11, 9) (1, 6, 2, 7, 9), (3, 8, 4, 10, 5, 11)	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11) (1, 3, 5, 2, 6), (4, 7, 9, 11, 8, 10) (1, 4, 2, 7, 11), (3, 8, 5, 9, 6, 10)
$C_5 \cup C_7$	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11, 12) (1, 3, 5, 2, 4), (6, 8, 10, 7, 12, 9, 11) (1, 6, 2, 7, 9), (3, 8, 4, 10, 12, 5, 11)	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11, 12) (1, 3, 5, 2, 4), (6, 8, 11, 7, 9, 12, 10) (1, 6, 2, 7, 10), (3, 8, 4, 9, 11, 5, 12)
$C_5 \cup C_8$	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11, 12, 13) (1, 3, 5, 2, 4), (6, 8, 10, 7, 11, 13, 9, 12) (1, 6, 2, 7, 9), (3, 8, 4, 11, 5, 12, 10, 13)	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11, 12, 13) (1, 3, 5, 2, 4), (6, 8, 10, 7, 12, 9, 13, 11) (1, 6, 2, 7, 9), (3, 8, 4, 11, 5, 12, 10, 13)
$C_5 \cup C_9$	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11, 12, 13, 14) (1, 3, 5, 2, 4), (6, 8, 10, 7, 9, 12, 14, 11, 13) (1, 6, 2, 7, 11), (3, 8, 4, 9, 13, 5, 12, 10, 14)	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11, 12, 13, 14) (1, 3, 5, 2, 4), (6, 8, 10, 7, 11, 13, 9, 14, 12) (1, 6, 2, 7, 9), (3, 8, 4, 11, 14, 5, 12, 10, 13)
$C_5 \cup C_{10}$	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11, 12, 13, 14, 15) (1, 3, 5, 2, 4), (6, 8, 10, 7, 9, 11, 13, 15, 12, 14) (1, 6, 2, 7, 11), (3, 8, 4, 12, 5, 13, 9, 14, 10, 15)	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11, 12, 13, 14, 15) (1, 3, 5, 2, 4), (6, 8, 10, 7, 9, 12, 14, 11, 15, 13) (1, 6, 2, 7, 11), (3, 8, 4, 9, 13, 5, 12, 15, 10, 14)
$C_6 \cup C_6$	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 11) (1, 4, 6, 2, 7, 9), (3, 10, 5, 11, 8, 12)	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12) (1, 3, 5, 2, 4, 7), (6, 9, 11, 8, 12, 10) (1, 4, 6, 2, 7, 11), (3, 8, 10, 5, 9, 12)
$C_6 \cup C_7$	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12, 13) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 11, 13) (1, 4, 6, 2, 7, 9), (3, 10, 13, 5, 11, 8, 12)	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12, 13) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 13, 11) (1, 4, 6, 2, 7, 9), (3, 10, 13, 5, 11, 8, 12)
$C_6 \cup C_8$	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 13, 11, 14) (1, 4, 6, 2, 7, 5), (3, 9, 11, 8, 12, 14, 10, 13)	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 14, 11, 13) (1, 4, 6, 2, 7, 5), (3, 8, 11, 9, 13, 10, 14, 12)
$C_6 \cup C_9$	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14, 15) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 13, 15, 11, 14) (1, 4, 6, 2, 7, 5), (3, 8, 11, 9, 14, 12, 15, 10, 13)	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14, 15) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 14, 11, 13, 15) (1, 4, 6, 2, 7, 5), (3, 8, 11, 9, 13, 10, 14, 12, 15)
$C_6 \cup C_{10}$	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14, 15, 16) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 11, 14, 16, 13, 15) (1, 4, 6, 2, 7, 5), (3, 8, 11, 13, 9, 14, 10, 15, 12, 16)	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14, 15, 16) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 13, 15, 11, 14, 16) (1, 4, 6, 2, 7, 5), (3, 8, 11, 9, 14, 12, 15, 10, 13, 16)
$2C_3 \cup C_4$	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10) (1, 4, 7), (2, 5, 8), (3, 9, 6, 10) (1, 5, 9), (2, 4, 10), (3, 7, 6, 8)	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10) (1, 4, 7), (2, 5, 9), (3, 8, 6, 10) (1, 5, 8), (2, 4, 10), (3, 7, 6, 9)
$2C_3 \cup C_5$	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11) (1, 4, 7), (2, 5, 8), (3, 9, 11, 6, 10) (1, 5, 9), (2, 4, 10), (3, 7, 6, 8, 11)	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11) (1, 4, 7), (2, 5, 9), (3, 8, 10, 6, 11) (1, 5, 8), (2, 4, 11), (3, 9, 6, 7, 10)

Table 1 cont.

2-factor	First 3-packing	Second 3-packing
$2C_3 \cup C_6$	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12) (1, 4, 7), (2, 5, 8), (3, 6, 10, 12, 9, 11) (1, 5, 9), (2, 4, 10), (3, 7, 6, 11, 8, 12)	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12) (1, 4, 7), (2, 5, 8), (3, 9, 11, 6, 10, 12) (1, 5, 9), (2, 4, 10), (3, 6, 12, 8, 11, 7)
$2C_3 \cup C_7$	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12, 13) (1, 4, 7), (2, 5, 8), (3, 6, 9, 11, 13, 10, 12) (1, 5, 9), (2, 4, 10), (3, 7, 11, 6, 12, 8, 13)	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12, 13) (1, 4, 7), (2, 5, 8), (3, 6, 10, 12, 9, 13, 11) (1, 5, 9), (2, 4, 10), (3, 7, 11, 6, 12, 8, 13)
$2C_3 \cup C_8$	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14) (1, 4, 7), (2, 5, 8), (3, 6, 9, 11, 13, 10, 12, 14) (1, 5, 9), (2, 4, 10), (3, 7, 11, 14, 6, 12, 8, 13)	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14) (1, 4, 7), (2, 5, 8), (3, 6, 9, 11, 13, 10, 14, 12) (1, 5, 9), (2, 4, 10), (3, 7, 11, 14, 6, 12, 8, 13)
$2C_3 \cup C_9$	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14, 15) (1, 4, 7), (2, 5, 8), (3, 6, 9, 11, 13, 10, 14, 12, 15) (1, 5, 9), (2, 4, 10), (3, 7, 6, 12, 8, 13, 15, 11, 14)	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14, 15) (1, 4, 7), (2, 5, 8), (3, 6, 9, 11, 13, 10, 15, 12, 14) (1, 5, 9), (2, 4, 10), (3, 7, 6, 12, 8, 14, 11, 15, 13)
$2C_3 \cup C_{10}$	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14, 15, 16) (1, 4, 7), (2, 5, 8), (3, 6, 9, 11, 13, 10, 14, 16, 12, 15) (1, 5, 9), (2, 4, 10), (3, 7, 6, 8, 12, 14, 11, 15, 13, 16)	(1, 2, 3), (4, 5, 6), (7, 8, 9, 10, 11, 12, 13, 14, 15, 16) (1, 4, 7), (2, 5, 8), (3, 6, 9, 11, 13, 10, 15, 12, 14, 16) (1, 5, 9), (2, 4, 10), (3, 7, 6, 11, 14, 8, 12, 16, 13, 15)
$C_3 \cup 2C_4$	(1, 2, 3), (4, 5, 6, 7), (8, 9, 10, 11) (1, 4, 6), (2, 5, 8, 10), (3, 7, 9, 11) (1, 5, 7), (2, 8, 3, 9), (4, 10, 6, 11)	(1, 2, 3), (4, 5, 6, 7), (8, 9, 10, 11) (1, 4, 6), (2, 5, 8, 10), (3, 9, 7, 11) (1, 5, 7), (2, 4, 9, 11), (3, 8, 6, 10)
$C_3 \cup C_4 \cup C_5$	(1, 2, 3), (4, 5, 6, 7), (8, 9, 10, 11, 12) (1, 4, 6), (2, 5, 3, 7), (8, 10, 12, 9, 11) (1, 5, 8), (2, 9, 4, 10), (3, 6, 11, 7, 12)	(1, 2, 3), (4, 5, 6, 7), (8, 9, 10, 11, 12) (1, 4, 6), (2, 5, 3, 8), (7, 10, 12, 9, 11) (1, 5, 7), (2, 9, 3, 10), (4, 8, 11, 6, 12)
$C_3 \cup C_4 \cup C_6$	(1, 2, 3), (4, 5, 6, 7), (8, 9, 10, 11, 12, 13) (1, 5, 7), (2, 4, 8, 12), (3, 9, 11, 6, 10, 13) (1, 4, 6), (2, 5, 3, 7), (8, 10, 12, 9, 13, 11)	(1, 2, 3), (4, 5, 6, 7), (8, 9, 10, 11, 12, 13) (1, 5, 7), (2, 8, 3, 11), (4, 9, 13, 6, 10, 12) (1, 4, 6), (2, 5, 3, 7), (8, 10, 13, 11, 9, 12)
$3C_4$	(1, 2, 3, 4), (5, 6, 7, 8), (9, 10, 11, 12) (1, 3, 5, 7), (2, 4, 9, 11), (6, 8, 10, 12) (1, 5, 2, 6), (3, 9, 7, 10), (4, 11, 8, 12)	(1, 2, 3, 4), (5, 6, 7, 8), (9, 10, 11, 12) (1, 3, 5, 7), (2, 4, 9, 11), (6, 10, 8, 12) (1, 5, 2, 6), (3, 9, 8, 11), (4, 7, 10, 12)
$2C_4 \cup C_5$	(1, 2, 3, 4), (5, 6, 7, 8), (9, 10, 11, 12, 13) (1, 3, 5, 7), (2, 4, 6, 8), (9, 11, 13, 10, 12) (1, 5, 2, 9), (3, 10, 6, 11), (4, 7, 12, 8, 13)	(1, 2, 3, 4), (5, 6, 7, 8), (9, 10, 11, 12, 13) (1, 3, 5, 7), (2, 4, 6, 8), (9, 11, 13, 10, 12) (1, 5, 2, 9), (3, 10, 7, 11), (4, 8, 12, 6, 13)
$2C_4 \cup C_6$	(1, 2, 3, 4), (5, 6, 7, 8), (9, 10, 11, 12, 13, 14) (1, 6, 9, 12), (4, 11, 8, 14), (2, 5, 3, 13, 10, 7) (1, 3, 6, 8), (2, 4, 5, 9), (7, 12, 10, 14, 11, 13)	(1, 2, 3, 4), (5, 6, 7, 8), (9, 10, 11, 12, 13, 14) (2, 12, 3, 13), (4, 6, 10, 7), (1, 5, 14, 11, 9, 8) (1, 3, 5, 7), (2, 4, 8, 10), (6, 11, 13, 9, 12, 14)
$C_4 \cup 2C_5$	(1, 2, 3, 4), (5, 6, 7, 8, 9), (10, 11, 12, 13, 14) (2, 9, 4, 14), (1, 3, 12, 8, 5), (6, 10, 7, 13, 11) (1, 6, 2, 7), (3, 5, 4, 8, 11), (9, 13, 10, 12, 14)	(1, 2, 3, 4), (5, 6, 7, 8, 9), (10, 11, 12, 13, 14) (1, 3, 9, 11), (2, 5, 14, 7, 12), (4, 6, 10, 8, 13) (1, 5, 3, 6), (2, 4, 7, 9, 14), (8, 11, 13, 10, 12)
$C_4 \cup C_5 \cup C_6$	(1, 2, 3, 4), (5, 6, 7, 8, 9), (10, 11, 12, 13, 14, 15) (3, 14, 8, 15), (1, 10, 2, 7, 13), (4, 5, 11, 6, 9, 12) (1, 3, 5, 7), (2, 4, 6, 8, 11), (9, 10, 13, 15, 12, 14)	(1, 2, 3, 4), (5, 6, 7, 8, 9), (10, 11, 12, 13, 14, 15) (2, 6, 12, 7), (1, 10, 14, 5, 13), (3, 9, 15, 8, 4, 11) (1, 3, 5, 7), (2, 4, 6, 8, 10), (9, 11, 13, 15, 12, 14)
$3C_5$	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10), (11, 12, 13, 14, 15) (1, 7, 11, 2, 14), (3, 5, 9, 6, 15), (4, 10, 12, 8, 13) (1, 3, 6, 2, 4), (5, 7, 9, 11, 8), (10, 13, 15, 12, 14)	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10), (11, 12, 13, 14, 15) (1, 7, 5, 2, 15), (3, 6, 13, 4, 8), (9, 12, 10, 11, 14) (1, 3, 5, 6, 4), (2, 7, 9, 11, 8), (10, 13, 15, 12, 14)
$2C_5 \cup C_6$	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10), (11, 12, 13, 14, 15, 16) (1, 4, 16, 10, 12), (2, 7, 9, 5, 8), (3, 14, 6, 13, 11, 15) (1, 3, 5, 2, 6), (4, 7, 10, 8, 11), (9, 12, 14, 16, 13, 15)	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10), (11, 12, 13, 14, 15, 16) (1, 10, 15, 6, 12), (2, 7, 4, 14, 16), (3, 5, 13, 9, 11, 8) (1, 3, 6, 2, 4), (5, 7, 9, 12, 8), (10, 14, 11, 15, 13, 16)
$C_5 \cup 2C_6$	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11), (12, 13, 14, 15, 16, 17) (4, 10, 14, 16, 13), (1, 8, 6, 15, 9, 17), (2, 7, 12, 3, 5, 11) (1, 3, 6, 2, 4), (5, 7, 9, 11, 8, 13), (10, 15, 17, 14, 12, 16)	(1, 2, 3, 4, 5), (6, 7, 8, 9, 10, 11), (12, 13, 14, 15, 16, 17) (1, 10, 2, 6, 15), (3, 7, 16, 14, 11, 9), (4, 8, 13, 17, 5, 12) (1, 4, 2, 5, 3), (6, 8, 10, 7, 9, 13), (11, 15, 17, 14, 12, 16)
$3C_6$	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12), (13, 14, 15, 16, 17, 18) (1, 5, 14, 7, 3, 13), (2, 12, 18, 10, 6, 16), (4, 8, 11, 17, 9, 15) (1, 3, 5, 2, 4, 7), (6, 8, 10, 12, 9, 11), (13, 15, 17, 14, 18, 16)	(1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12), (13, 14, 15, 16, 17, 18) (1, 4, 2, 17, 10, 12), (3, 5, 8, 18, 16, 13), (6, 9, 11, 15, 7, 14) (1, 3, 6, 2, 5, 7), (4, 8, 10, 13, 9, 12), (11, 16, 14, 17, 15, 18)
$4C_3$	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12) (1, 5, 8), (2, 4, 12), (3, 9, 10), (6, 7, 11) (1, 4, 7), (2, 5, 10), (3, 8, 11), (6, 9, 12)	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12) (1, 5, 8), (2, 7, 12), (3, 6, 10), (4, 9, 11) (1, 4, 7), (2, 5, 10), (3, 8, 11), (6, 9, 12)
$3C_3 \cup C_4$	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12, 13) (1, 5, 9), (2, 4, 10), (3, 6, 11), (7, 12, 8, 13) (1, 4, 7), (2, 5, 8), (3, 10, 12), (6, 9, 11, 13)	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12, 13) (1, 5, 9), (2, 4, 10), (3, 11, 13), (6, 7, 12, 8) (1, 4, 7), (2, 5, 8), (3, 10, 12), (6, 11, 9, 13)
$3C_3 \cup C_5$	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12, 13, 14) (2, 9, 10), (3, 6, 12), (4, 7, 14), (1, 5, 13, 8, 11) (1, 4, 8), (2, 5, 7), (3, 9, 11), (6, 13, 10, 12, 14)	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12, 13, 14) (1, 4, 12), (2, 7, 10), (5, 9, 11), (3, 8, 14, 6, 13) (1, 5, 7), (2, 4, 8), (3, 6, 9), (10, 12, 14, 11, 13)
$3C_3 \cup C_6$	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12, 13, 14, 15) (1, 9, 11), (3, 5, 7), (8, 12, 15), (2, 4, 14, 6, 10, 13) (1, 4, 7), (2, 5, 8), (3, 6, 12), (9, 10, 14, 11, 13, 15)	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12, 13, 14, 15) (2, 4, 8), (6, 13, 15), (10, 12, 14), (1, 9, 3, 7, 5, 11) (1, 4, 7), (2, 5, 9), (3, 6, 14), (8, 10, 13, 11, 15, 12)
$5C_3$	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12), (13, 14, 15) (1, 4, 7), (2, 10, 13), (3, 6, 14), (5, 8, 11), (9, 12, 15) (1, 5, 9), (2, 4, 8), (3, 10, 15), (6, 11, 13), (7, 12, 14)	(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12), (13, 14, 15) (1, 4, 7), (2, 5, 8), (3, 12, 15), (6, 10, 13), (9, 11, 14) (1, 5, 9), (2, 4, 10), (3, 6, 14), (7, 11, 15), (8, 12, 13)

Igor Grzelec (corresponding author)
grzelec@agh.edu.pl
https://orcid.org/0000-0002-1011-535X

AGH University of Krakow
Faculty of Applied Mathematics
Department of Discrete Mathematics
al. A. Mickiewicza 30, 30–059 Kraków, Poland

Tomáš Madaras
tomas.madaras@upjs.sk
https://orcid.org/0000-0003-4565-043X

P.J. Šafárik University
Institute of Mathematics, Faculty of Science
Jesenná 5, 041 54 Košice, Slovak Republic

Alfréd Onderko
alfred.onderko@upjs.sk
https://orcid.org/0000-0003-4811-3955

P.J. Šafárik University
Institute of Mathematics, Faculty of Science
Jesenná 5, 041 54 Košice, Slovak Republic

Received: March 18, 2024.
Revised: September 21, 2024.
Accepted: October 7, 2024.