Stability and scheduling in wireless networks with local pooling

Dariusz Dereniowski[*] Wieslaw Kubiak[†] Bernard Ries[‡] Yori Zwols[§]

October 15, 2011

Abstract

In a single-hop traffic model in a wireless network with primary interference constraints, a matching needs to be found in each time slot. This matching then determines the links used for transmission in that time slot. The matchings however need to be selected so that any given link occurs in them sufficiently often to keep up with the packet arrival rate on that link, because otherwise the backlog on that link grows to infinity. We aim at designing an efficient test of network stability for a class of networks which generalizes the recently defined class of OLoP(*Overall Local Pooling*) networks. The test takes the network topology (a graph) and the link arrival rates as an input and determines, in linear time, whether or not there exists a sequence of matchings that keeps up with the arrival rates. If the answer is affirmative we show how to construct, in quadratic time, the sequence with the number of distinct matchings that grows linearly with the number of vertices in the graph. Finally, we focus on the shortest possible sequence. We prove that the shortest sequence length does not exceed the least common denominator of the arrival rates for OLoP networks, and conjecture that it does not exceed twice the least common denominator of arrival rates for general networks. We then show that this conjecture holds for all graphs with at most 10 vertices.

Keywords: greedy maximal scheduling, edge coloring, nearly bipartite graphs, scheduling algorithms, throughput maximization, wireless networks

[*]Gdańsk University of Technology, Gdańsk, Poland, deren@eti.pg.gda.pl
[†]Memorial University, St. John's, Canada, wkubiak@mun.ca
[‡]Université Paris Dauphine, Paris, France, bernard.ries@dauphine.fr
[§]McGill University, Montreal, Canada, yori.zwols@mail.mcgill.ca

1 Introduction

We begin with introducing a wireless networking problem and reviewing recent results in this field. The problem provides the main source of motivation for the stability and scheduling problems we study in this paper.

Consider a graph $G = (V, E)$, in which the vertices in set V represent agents (i.e., transmitters and receivers) in a communication network, and $E \subseteq \{ij : i, j \in V, i \neq j\}$ is a set of wireless connections representing pairs of agents between which data flow can occur. At each vertex $v \in V$ of the network, information packets are received over time and these packets must be transmitted to their destinations, which correspond in our model to the neighbors of that vertex v (such a model is called *single-hop*). We assume that time is slotted and that packets are of equal size, each packet requiring one time slot of service across a link. A stochastic queue is associated with each edge in the network, representing the packets waiting to be transmitted on this link. We assume that the stochastic arrivals to edge ij have long term rates λ_{ij} and are independent of each other. We denote by $\boldsymbol{\lambda}$ the vector of the arrival rates λ_{ij} for every edge ij.

An important issue in operating a wireless communication network is that two connections might interfere with each other. We focus on the simplest interference model that can be found in the literature, which is known as the *primary interference model*. This model states that two connections interfere with each other if and only if the corresponding edges share a vertex in G. Thus, at every time slot, the set of connections that are activated should form a matching.

A *scheduling algorithm* selects a set of edges to activate at each time slot, and transmits packets on those edges. The goal is to find a scheduling algorithm that, informally speaking, keeps the sizes of the queues from growing unboundedly when this algorithm is adopted. Clearly, if the values of the λ_{ij}'s are chosen sufficiently large, no algorithm can attain this. Therefore, usually, one defines the *stability region* Λ^* of a graph G (with respect to rates λ) as

$$\Lambda^* = \left\{ \boldsymbol{\lambda} \in \mathbb{R}^{E(G)} : \boldsymbol{\lambda} < \mathbf{u} \text{ for some } \mathbf{u} \in \mathbf{conv}(\mathcal{M}_G) \right\},$$

where $\mathcal{M}_G$ is the set of all matchings in G and $\mathbf{conv}(\mathcal{M}_G)$ is the convex hull of the characteristic (0-1) vectors of the elements of $\mathcal{M}_G$ (and the '$<$' sign is componentwise). It should be clear that, for any λ that is not an element of the closure of Λ^*, there is no algorithm that prevents the queues from growing unboundedly. On the other hand, one may ask whether there exists a scheduling algorithm that keeps the queues from growing unboundedly when $\lambda \in \Lambda^*$. Formally, this is equivalent to the condition that the Markov chain that represents the evolution of the queues is positive recurrent (i.e., all its states are positive recurrent) for all arrival rates $\boldsymbol{\lambda} \in \Lambda^*$, using the scheduling algorithm. If this condition is satisfied, then we say that the scheduling algorithm *achieves* 100% *throughput on* G. For more details regarding the queue evolution process under this model, see [3, 4, 14].

A good first choice for a scheduling algorithm turns out to be the *Maximum Weight Matching algorithm (MWM)* that selects, in each time slot, a maximum weight matching in G, where the weights of the edges are given by the current queue lengths. It was shown in [22] that MWM achieves 100% throughput on any graph G. However, it is not a tractable algorithm in many situations because to find an optimal solution it needs centralized computing and full knowledge of both the network topology and all queue lengths at every time slot. Hence, there has been an increasing interest in simple and potentially distributed algorithms. One example of such an algorithm is known as the *Greedy Maximal Scheduling (GMS) algorithm* [12, 17]. This distributed algorithm effectively selects the set of links served in a greedy fashion according to the queue lengths at these links (i.e., GMS greedily selects a maximal weight matching). A drawback of using this algorithm is that, in general, it does not achieve 100% throughput for every graph G. However, [4] gave a sufficient condition on network graphs (in the primary interference model) for which the GMS algorithm does achieve 100% throughput. We say that a graph G is *OLoP* (*OLoP* stands for *Overall Local Pooling*) if, for every subgraph G' of G, there exists a function $w : E(G') \to [0, 1]$ (that depends on G') such that every inclusion-wise maximal matching M in G' satisfies $\sum_{e \in M} w(e) = 1$. The sufficient condition from [4] is the following.

Theorem. [4] *GMS achieves* 100% *throughput on every OLoP graph* G.

In [2], a complete structural characterization of all OLoP graphs was given. We repeat this characterization in Section 4.1 for completeness.

This paper focuses on the following four problems that are studied for general graphs and GOLoP graphs (*GOLoP* stands for *Generalized Overall Local Pooling*) in particular, the latter being a generalization of OLoP graphs which we will introduce in Section 4.1.

Problem: MATCH(G, r)

Input: An undirected, simple graph $G = (V, E)$, a rate function r that assigns to each edge $e \in E$ a rational rate $0 < r(e) \leq 1$.

Question: Does there exist a sequence of matchings $M_1, M_2, \ldots$ in G such that

$$\liminf_{n\to\infty} \frac{|\{i : e \in M_i, i = 1, 2, \ldots, n\}|}{n} \geq r(e) \quad \text{for all } e \in E? \tag{1}$$

The problem MATCH(G, r) falls in the general framework of a problem earlier studied by Grötschel et al. in [10], and by Hajek and Sasaki in [11]. The problem is known as 'scheduling links to satisfy link demand', where the allowable links being active in a network at any given time must form a matching. [10] gives a polynomial time algorithm for the problem, and [11] improves the result by giving a $O(|E| \cdot |V|^5)$-time algorithm that finds a schedule of minimum length which meets all link demands. (The length of an optimal, i.e. minimum length, schedule can be determined in $O(|V|^5)$-time [11].) This paper focuses on these problems restricted to the class of GOLoP graphs. Within this setting, we obtain a linear-time algorithm for answering MATCH(G, r).

Next, we are interested in the following finite version of MATCH(G, r):

Problem: K-MATCH(G, r, k)

Input: An undirected, simple graph $G = (V, E)$, a rate function r that assigns to each edge $e \in E$ a rational rate $0 < r(e) \leq 1$, and a positive integer k.

Question: Are there k matchings $M_1, \ldots, M_k$ in G such that

$$|\{i : e \in M_i\}| \geq kr(e) \quad \text{for all } e \in E? \tag{2}$$

Although this problem turns out to be $\mathcal{NP}$-complete on general graphs (see Section 3), we obtain a linear-time algorithm for solving this problem with input restricted to GOLoP graphs. Given an affirmative answer for K-MATCH(G, r, k), we naturally consider the following problem.

Problem: FIND-MATCH(G, r, k)

Input: An undirected, simple graph $G = (V, E)$, a rate function r that assigns to each edge $e \in E$ a rational rate $0 < r(e) \leq 1$, and a positive integer k.

Question: Find k matchings $M_1, \ldots, M_k$ in G such that

$$|\{i : e \in M_i\}| \geq kr(e) \quad \text{for all } e \in E.$$

We will show that FIND-MATCH(G, r, k) can be solved in $O(|V(G)|^2)$ time on GOLoP graphs. Finally, we will use the solutions to the latter two problems to find in pseudopolynomial time for the following problem:

Problem: MIN-MATCH(G, r)

Input: An undirected, simple graph $G = (V, E)$, a rate function r that assigns to each edge $e \in E$ a rational rate $0 < r(e) \leq 1$.

Question: Find the smallest k such that there exist matchings $M_1, \ldots, M_k$ in G with

$$|\{i : e \in M_i\}| \geq kr(e) \quad \text{for all } e \in E.$$

The paper is organized as follows. Section 1.1 introduces notation and definitions used in the paper. Section 2 explains the relationship between the problem MATCH(G, r) and fractional edge-coloring, and the relationship between the problems K-MATCH(G, r, k) and FIND-MATCH(G, r, k) and edge-coloring. The computational complexity of MATCH(G, r) and K-MATCH(G, r, k) is briefly analyzed in Section 3. Section 4.1 gives characterizations of OLoP and GOLoP graphs used throughout the remaining sections. Sections 4.2, 4.3, and 4.4 give linear-time solutions to K-MATCH(G, r, k) and MATCH(G, r), and find solutions that use only $O(|V(G)|)$ distinct matchings for FIND-MATCH(G, r, k) in $O(|V(G)|^2)$ time, respectively, for GOLoP graphs. Section 5.1 shows that, for OLoP graphs, the least common denominator d for the rate function r is an upper bound for the length of the shortest schedule, given that the problem K-MATCH(G, r, k) has an affirmative answer for some k. Furthermore, following a conjecture of Seymour [21], the section conjectures that this bound does not exceed $2d$ for general graphs, and it proves the conjecture for all graphs with $|V(G)| \leq 10$. Section 5.2 proves an upper bound for GOLoP graphs. Section 5.3 gives a pseudopolynomial time algorithm for the shortest schedule for GOLoP graphs. The algorithm's running time depends on the upper bounds found in Sections 5.1 and 5.2. Finally, Section 6 concludes the paper and presents some open questions.

1.1 Preliminaries

All graphs in this paper are connected and finite. Let $G = (V, E)$ be a graph. For a vertex $v \in V$, let $N_G(v)$ denote the set of vertices in G that are adjacent to v, i.e., the neighbors of v. $N_G(v)$ is called the *neighborhood* of vertex v. Whenever it is clear from the context what G is, we will drop the subscripts and write $N(v) = N_G(v)$. The *degree* of a vertex v, denoted by $\deg(v)$, is the number of neighbors of v, i.e., $\deg(v) = |N(v)|$. The set of edges incident to a vertex v will be denoted by $\delta(v)$. $\Delta(G)$ denotes the *maximum degree* of G, i.e., $\Delta(G) = \max_{v\in V}\{\deg(v)\}$. For $X \subseteq V$ we denote by $G[X]$ the subgraph induced by X. We write $G - v$ for the subgraph obtained from G by deleting a vertex v. Similarly, for $X \subseteq V$, we denote by $G - X$ the subgraph of G obtained by deleting the set X, i.e., $G - X = G[V \setminus X]$. A *matching* in a graph $G = (V, E)$ is a set of pairwise non-adjacent edges. The set of all matchings in G is denoted by $\mathcal{M}_G$ and the set of all matchings in G containing an edge $e \in E(G)$ is denoted by $\mathcal{M}_G(e)$. For $u \in V(G)$, let $\mathcal{M}_G(u)$ denote the set of matchings that contain an edge incident with u, and for $v \in V(G)$, $v \neq u$, let $\mathcal{M}_G(u, v) = \mathcal{M}_G(u) \cap \mathcal{M}_G(v)$. If $M \in \mathcal{M}_G(u)$, we say that the matching M *saturates* u.

Let $G = (V, E)$ be a connected graph. We call $x \in V$ a *cut-vertex of* G if $G-x$ is not connected. We call a maximal connected induced subgraph B of G such that B has no cut-vertex a *block of* G. Let $B_1, B_2, \ldots, B_q$ be the blocks of G. We call the collection $\{B_1, B_2, \ldots, B_q\}$ the *block decomposition of* G. It is known that the block decomposition is unique and that $E(B_1), E(B_2), \ldots, E(B_q)$ (where $E(B_i)$ denotes the set of edges in B_i, $i = 1, \ldots, q$) form a partition of E (see for instance [23]). Furthermore, the vertex sets $V(B_i)$ and $V(B_j)$ of every two blocks B_i and B_j, $i, j = 1, \ldots, q$, $i \neq j$, intersect in at most one vertex and this vertex is a cut-vertex of G. Block decompositions give a tree-like decomposition of a graph in the following sense. Construct the *block-cutpoint graph of* G by keeping the cut-nodes of G and replacing each block B_i of G by a node b_i. Make each cut-node v adjacent to b_i if and only if $v \in V(B_i)$. It is known that the block-cutpoint graph of G forms a tree (e.g., [23]). With this tree-like structure in mind, we say that a block B_i is a *leaf block* if it contains at most one cut-node of G. Clearly, if $q \geq 2$, then $\{B_i\}_{i=1}^q$ contains at least two leaf blocks.

A *multigraph* H is a pair (G, mp) where G is a graph (with no parallel edges) and $\mathrm{mp} : E(G) \to \mathbb{Z}_+$ is a function. The value of $\mathrm{mp}(e)$ is the *multiplicity* of edge e in the multigraph H. We define $V(H) = V(G)$, $E(H) = E(G)$, and $\mathcal{M}_H = \mathcal{M}_G$. (Thus, to clarify, $\mathcal{M}_H$ does not contain two matchings that differ only in the choice of parallel edges.) For $v \in V(H)$, we define $\deg_H(v) = \sum_{e\in\delta(v)} \mathrm{mp}(e)$. Moreover, $\Delta(H) = \max_{v\in V(H)} \deg(v)$ denotes the maximum degree of H. An *edge coloring* of a multigraph H is a mapping $c :\ E(H) \to 2^{\mathbb{Z}_+}$ such that $|c(e)| = \mathrm{mp}(e)$ for all $e \in E(H)$ and if $e_1, e_2 \in E(H)$ share a vertex, then $c(e_1) \cap c(e_2) = \emptyset$. Let $e_{max} = \max\{z | z \in c(e)\}$. If for an edge-coloring c we have $\max_{e\in E(H)}\{e_{max}\} \leq k$ for all $e \in E$, then we call c a *k-edge-coloring* of H. The smallest integer k such that H admits a k-edge-coloring is called the *chromatic index* of H and is denoted by $\chi'(H)$. It is well-known that $\chi'(H)$ can be written as the optimal value of the following integer linear program:

$$\begin{aligned} \chi'(H) = \chi'(G, \mathrm{mp}) = \underset{\mathbf{x}\in\mathbb{Z}_+^{\mathcal{M}_G}}{\text{minimize}} \quad & \sum_{M\in\mathcal{M}_H} x(M) \\ \text{subject to} \quad & \sum_{M\in\mathcal{M}_H(e)} x(M) = \mathrm{mp}(e) \text{ for all } e \in E(H). \end{aligned} \tag{3}$$

A *fractional edge coloring* of a graph G is a mapping $f :\ \mathcal{M}_G \to \mathbb{R}_+$ such that $\sum_{M\in\mathcal{M}_G(e)} f(M) = 1$. If for a fractional edge coloring f we have $\sum_{M\in\mathcal{M}_G(e)} f(M) \leq k$ for every edge e, we call f a *fractional k-edge coloring.* The smallest integer k such that G admits a fractional k-edge coloring is called the *fractional chromatic index* of G and is denoted by $\chi'_f(G)$.

A *rate function* (for a graph G) is a function $r : E(G) \to \mathbb{Q} \cap (0, 1]$. Similar to the degree of a vertex, for any $v \in V(G)$, we will write $r(x) = \sum_{e \in \delta(x)} r(e)$. We will think of rate functions as a continuous versions of multiplicity functions mp. With this in mind, we introduce the following continuous version of the fractional chromatic index. For a graph G and a weight function $w : E(G) \to \mathbb{R}_+$ (we will usually take w to be either a multiplicity function mp, or a rate function r), define

$$\begin{aligned} \chi'_f(G, w) = \quad & \underset{\mathbf{x} \in \mathbb{R}_+^{\mathcal{M}_G}}{\text{minimize}} & & \sum_{M \in \mathcal{M}_G} x(M) \\ & \text{subject to} & & \sum_{M \in \mathcal{M}_G(e)} x(M) = w(e) \text{ for all } e \in E(G). \end{aligned} \tag{4}$$

Given a multigraph $H = (G, \text{mp})$, define

$$t(H) = \max \left\{ \frac{2|E(H')|}{|V(H')| - 1} : H' \text{ is an induced subgraph of } H, |V(H')| \text{ is odd}, |V(H')| \geq 3 \right\}.$$

We have the following fundamental result for the fractional chromatic index of any multigraph.

Theorem 1.1. (Edmonds [5]) $\chi'_f(H) = \max\{\Delta(H), t(H)\}$, *for every multigraph* H.

2 Relationship with edge-coloring

For a given rate function r and an integer k, we will write $\lceil kr \rceil$ for the function $e \mapsto \lceil kr(e) \rceil$. We have the following equivalence.

(2.1) *Let G be a graph, let r be a rate function for G, and let k be an integer. The answer to* K-MATCH(G, r, k) *is* YES *if and only if* $\chi'(G, \lceil kr \rceil) \leq k$.

Proof. If the answer to K-MATCH(G, r, k) is YES, then there exist k matchings $M_1, \ldots, M_k$ in G such that (2) is satisfied. Then, clearly,

$$|\{i : e \in M_i\}| \geq \lceil kr(e) \rceil \quad \text{for all } e \in E$$

and thus there exist k matchings $M'_1, \ldots, M'_k$ in G such that

$$|\{i : e \in M'_i\}| = \lceil kr(e) \rceil \quad \text{for all } e \in E.$$

We obtain a feasible k-edge-coloring of the multigraph $(G, \lceil kr \rceil)$ by setting $c(e) = \{i : e \in M'_i\}$.

Suppose now that there exists a k-edge-coloring c of the multigraph $(G, \lceil kr \rceil)$. Define $M_i = \{e \in E : i \in c(e)\}$ for $i = 1, \ldots, k$. Clearly, each M_i defines a matching and for each edge $e \in E$ we have $|\{M_i : e \in M_i\}| = \text{mp}(e) = \lceil kr(e) \rceil$. Thus the matchings $M_1, \ldots, M_k$ satisfy (2). This proves (2.1). ■

We also have the following equivalence:

(2.2) *Let G be a graph, let r be a rate function for G, and let k be an integer. The following three statements are equivalent:*

(i) $\chi'_f(G,r) \le 1$;

(ii) *The answer to* MATCH(G,r) *is* YES*;*

(iii) *The answer to* K-MATCH(G,r,k) *is* YES *for some* k*.*

Proof. (i) $\implies$ (iii): Suppose that $\chi'_f(G,r) \le 1$. Consider problem (4) corresponding to $\chi'_f(G,r)$. Since the parameters of this linear program are all rational, there exists a solution $p(M)$, $M \in \mathcal{M}_G$ to (4) such that $\sum_{M\in\mathcal{M}_G} p(M) \le 1$ and $p(M)$ is rational for every $M \in \mathcal{M}_G$. Let d be the least common denominator of $p(M)$, $M \in \mathcal{M}_G$. Define $q(M) = dp(M)$ and let $T = \sum_{M\in\mathcal{M}_G} q(M)$. Then, $q(M)$ is integral for all $M \in \mathcal{M}_G$ and so is T. Notice that $T = d\chi'_f(G,r) \le d$. Let $\{M_i\}_{i=1}^d$ be such that $M_i = M$ for exactly $q(M)$ values of i for each $M \in \mathcal{M}_G$ (filling up with $\emptyset$ matchings if necessary). Now,

$$|\{i : e \in M_i\}| = \sum_{M\in\mathcal{M}_G(e)} q(M) = d \sum_{M\in\mathcal{M}_G(e)} p(M) = dr(e),$$

as required. This implies that (iii) holds with $k = d$.

(iii) $\implies$ (ii): Let $\{M_i\}_{i=1}^k$ be such that $|\{i : e \in M_i, i = 1,2,\ldots,k\}| \ge kr(e)$ for all $e \in E(G)$. Let $\{M'_i\}_{i=1}^\infty$ be the sequence constructed by infinitely repeating the sequence $\{M_i\}_{i=1}^k$. Then, $\lim_{n\to\infty} \frac{|\{t\,:\,e\in M'_i,\ i=1,2,\ldots,n\}|}{n} = \frac{|\{i\,:\,e\in M_i,\ i=1,2,\ldots,k\}|}{k} \ge r(e)$ for every $e \in E$. Thus, the answer to MATCH(G,r) is YES, proving that (ii) holds.

(ii) $\implies$ (i): Suppose that the answer to MATCH(G,r) is YES. Let $\{M_i\}_{i=1}^\infty$ be a sequence of matchings satisfying (1). Let $e \in E$. Since every bounded sequence $\{x_n\} \subseteq \mathbb{R}$ has a subsequence that converges to $\liminf_{n\to\infty} x_n$, we can replace $\{M_i\}_{i=1}^\infty$ by a subsequence $\{M_{t_j}\}_{j=1}^\infty$ such that

$$\lim_{n\to\infty} \frac{|\{i_j : e \in M_{i_j}, j = 1,2,\ldots,n\}|}{n} = \liminf_{n\to\infty} \frac{|\{i : e \in M_i, i = 1,2,\ldots,n\}|}{n} \ge r(e).$$

Thus, we may, since $|E|$ is finite and by iteratively taking subsequences in this fashion, assume that for all $e \in E$

$$\lambda(e) := \lim_{n\to\infty} \frac{|\{i : e \in M_i, i = 1,2,\ldots,n\}|}{n} = \liminf_{n\to\infty} \frac{|\{i : e \in M_i, i = 1,2,\ldots,n\}|}{n} \ge r(e).$$

By replacing some matchings M_i by subsets $M'_i \subseteq M_i$ if necessary, we may assume that $\lambda(e) = r(e)$ for all $e \in E$. Likewise, we may assume that

$$p(M) := \lim_{n\to\infty} \frac{|\{i : M_i = M, i = 1,2,\ldots,n\}|}{n} = \liminf_{n\to\infty} \frac{|\{i : M_i = M, i = 1,2,\ldots,n\}|}{n}.$$

Clearly, we have $\sum_{M\in\mathcal{M}_G} p(M) = 1$. Now, $\{p(M)\}$ is a solution to (4) with objective value 1, thus proving that $\chi'_f(G,r) \le 1$. This proves the lemma. ■

3 Complexity of MATCH(G,r) and K-MATCH(G,r,k) for general graphs

In this section we will consider the problems MATCH(G,r) and K-MATCH(G,r,k) from a complexity point of view. For the former problem, we show that it is polynomial-time solvable. Though this was already shown in [11], we present here a new algorithm that improves the complexity of the one given in [11], where a $O(|V(G)|^5)$-time algorithm has been given.

(3.1) *The problem* $\textsc{Match}(G, r)$ *can be solved in* $O(|V(G)|^4)$ *time.*

Proof. Let $\mathcal{P}$ be the matching polyhedron corresponding to G (as defined by Edmonds [5]), i.e. $\mathcal{P} = \mathbf{conv}\,\{\mathbf{e}_M : M \in \mathcal{M}_G\}$, where $\mathbf{e}_M$ is the $|E(G)|$-dimensional 0-1 characteristic vector of M. We claim that $\chi'_f(G, r) \leq 1$ if and only if $r \in \mathcal{P}$. This is sufficient due to (2.2). To prove the claim, suppose that $r \in \mathcal{P}$. Then, $r = \sum_{M \in \mathcal{M}_G} p(M)\mathbf{e}_M$ for some $p : \mathcal{M}_G \to [0, 1]$ with $\sum_{M \in \mathcal{M}_G} p(M) = 1$. It follows that $\{p(M)\}$ is a feasible solution to (4) ensuring that $\chi'_f(G, r) \leq 1$. Conversely, suppose that $\chi'_f(G, r) \leq 1$. Let $\{p(M)\}$ be a solution to (4) such that $\sum_{M \in \mathcal{M}_G} p(M) = 1$. Such a solution can always be obtained by sufficiently increasing $p(\emptyset)$ if needed. By (4), we have $\sum_{M \in \mathcal{M}_G(e)} p(M) = r(e)$ for all $e \in E$. Thus, $r = \sum_{M \in \mathcal{M}_G} p(M)\mathbf{e}_M$, which shows that $r \in \mathcal{P}$. This proves the claim. Finally, the test whether $r \in \mathcal{P}$ can be done in polynomial time using the Padberg-Rao [18] separation algorithm. Moreover, Letchford, Reinelt and Theis [16] proved that the test can be done in $O(|V(G)|^4)$ time. ■

Although the complexity of $\textsc{Match}(G, r)$ is $O(|V(G)|^4)$ for general graphs, we will see in Section 4.3 that $\textsc{Match}(G, r)$ can be solved in linear time if G is a GOLoP graph (see 4.1 for the definition of a GOLoP graph). For the K-$\textsc{Match}(G, r, k)$ problem, we show the following using (2.1).

(3.2) K-$\textsc{Match}(G, r, k)$ *is* $\mathcal{NP}$*-complete in the strong sense.*

Proof. We will use a transformation from the k-edge coloring problem which is known to be $\mathcal{NP}$-complete (see [13]). Consider the k-edge coloring problem on a graph $G = (V, E)$. We define a rate function r on E such that $r(e) = \frac{1}{k}$ for each edge $e \in E$. Thus we obtain an instance of the K-$\textsc{Match}(G, r, k)$ problem. Now using (2.1), we immediately conclude that both problems are equivalent and hence K-$\textsc{Match}(G, r, k)$ is $\mathcal{NP}$-complete in the strong sense. ■

4 GOLoP graphs

In this section we first concentrate on the class of graphs for which the *Greedy Maximal Scheduling* (GMS) algorithm is optimal, that is, it achieves 100% throughput. This class of graphs has been characterized in [2]; as already mentioned in Section 1, these graphs are called *OLoP* graphs. Later in this section we will define another class of graphs which represents a generalization of OLoP graphs and which will be called GOLoP. Given a GOLoP graph $G = (V, E)$ with a rate function r and a positive integer k, we are interested in solving $\textsc{Match}(G, r)$, K-$\textsc{Match}(G, r, k)$ and $\textsc{Find-Match}(G, r, k)$.

4.1 Characterization of OLoP graphs; GOLoP graphs

We start with a characterization of OLoP graphs (see [2]). We will do this in terms of the block decomposition (see Section 1.1). It turns out that the block decomposition of an OLoP graph is relatively simple in the sense that there are only two types of blocks. The types are defined by the following two families of graphs.

$\mathcal{B}_1$: Construct $\mathcal{B}_1$ as follows. Let H be a graph with $V(H) = \{c_1, c_2, \ldots, c_k\}$, with $k \in \{5, 7\}$, such that

1. c_1-c_2-$\cdots$-c_k-c_1 is a cycle;
2. if $k = 5$, then the other adjacencies are arbitrary; if $k = 7$, then all other pairs are non-adjacent, except possibly $\{c_1, c_4\}$, $\{c_1, c_5\}$ and $\{c_4, c_7\}$.

Then, $H \in \mathcal{B}_1$.

Now iteratively perform the following operation. Let $H' \in \mathcal{B}_1$ and let $x \in V(H')$ with $\deg(x) = 2$. Construct H'' from H' by adding a vertex x' such that $N(x') = N(x)$. x' is called a *non-adjacent clone* of x. Then, $H'' \in \mathcal{B}_1$. We say that a graph is *of the* $\mathcal{B}_1$ *type* if it is isomorphic to a graph in $\mathcal{B}_1$.

$\mathcal{B}_2$: Let $\mathcal{B}_2 = \{K_2, K_3, K_4\} \cup \left\{K_{2,t}, K_{2,t}^+ : t \geq 2\right\}$, where $K_{2,t}^+$ is constructed from $K_{2,t}$ by adding an edge between the two vertices on the side of the bipartition that has cardinality 2. We say that a graph is *of the* $\mathcal{B}_2$ *type*, if it is isomorphic to a graph in $\mathcal{B}_2$.

In simple words, graphs of the $\mathcal{B}_1$ type are constructed as follows. Starting with a cycle of length five or seven. Then we may add some additional edges between vertex of the cycle, subject to some constraints. Finally, we may iteratively take a vertex x of degree 2 and add a non-adjacent clone x' of x. The following result characterizes OLoP graphs.

(4.1) ([2]) *Let* $G = (V, E)$ *be a graph and let* $\{B_1, \ldots, B_q\}$ *be its block decomposition.* G *is an* OLoP *graph if and only if at most one block of* G *is of the* $\mathcal{B}_1$ *type and all other blocks are of the* $\mathcal{B}_2$ *type.*

It follows from (4.1) that OLoP graphs can be constructed by starting with a block that is either of the $\mathcal{B}_1$ or of the $\mathcal{B}_2$ type, and then iteratively adding a block of the $\mathcal{B}_2$ type by 'glueing' it on an arbitrary vertex.

This motivates the following definition of a generalized OLoP graph. Let $b \geq 1$ be an integer. A graph G is called GOLoP(b) if every block of G can be obtained from a connected graph on at most b vertices by iteratively non-adjacent cloning a vertex of degree two. We say that a multigraph $H = (G, \mathrm{mp})$ is GOLoP(b) if the graph G is GOLoP(b). It is not hard to see that every OLoP graph is also a GOLoP(7) graph.

To deal with GOLoP graphs, we will frequently use the following notation. Let G be a GOLoP graph. Let $C_1, \ldots, C_p$ be maximal sets of vertices of degree two (in G) such that $|C_i| \geq 2$ and all vertices in set C_i have the same two neighbors u_i and v_i. We refere to these sets as non-adjecent clones in G. Choose p maximal as well. Consider the auxiliary graph G' constructed from $G - \bigcup_{i=1}^{p} C_i$ by adding new vertices $a_1, \ldots, a_p$ such that, for $i \in [p]$, a_i is adjacent to precisely u_i and v_i, and by adding new edges $u_i v_i$ for all $i \in [p]$. Let $W = \{a_1, \ldots, a_p\}$. We call the pair (G', W) the *collapsed graph* associated with G. For $i \in [p]$, let $F_i = H[C_i \cup \{u_i, v_i\}]$. See Figure 1.

It was shown in [2] that OLoP graphs have $O(|V(G)|)$ edges. The proof of this result generalizes easily to the setting of GOLoP(b). We include the generalization for completeness:

(4.2) *Let* b *be a fixed integer and let* G *be a* GOLoP(b) *graph. Then,* $|E(G)| = O(|V(G)|)$*.*

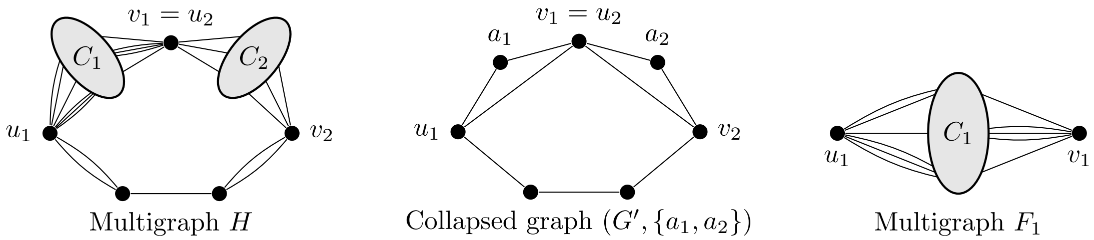

Figure 1: The multigraph H and the sets C_i (left), the corresponding collapsed graph (G', W) (middle), and one of the multigraphs F_1, F_2 (right). In this figure, we added multiple edges to represent the values of the function mp.

Proof. We may assume that G is connected, because otherwise the lemma follows from considering each connected component of G. First, let B be a block of a GOLoP(b) graph. Let B' be the collapsed graph of B. Then, since $|V(B)| \leq b$, we have $|E(B)| \leq \binom{b}{2}$. Since the degree of every clone is exactly two, there are at most $|V(B)| - 2$ non-adjacent clones in B. It follows that $|E(B)| \leq \binom{b}{2} + 2(|V(B)| - 2)$.

Now let G be a GOLoP(b) graph and let q be the number of blocks of the block-decomposition of G. We will prove by induction the stronger statement that $|E(G)| \leq q\binom{b}{2} + 2|V(G)|$. This will imply the lemma because $q = O(|V(G)|)$ and b is a constant. So consider G and let $\{B_1, B_2, \ldots, B_q\}$ be the block decomposition of G. We prove the lemma by induction on q. If $q = 1$, then G has exactly one block and the result follows from the above. So we may assume that $q \geq 2$. It follows that G has at least one leaf-block B. Let x be the unique cut-vertex of G in $V(B)$. We have $|E(B)| \leq \binom{b}{2} + 2(|V(B)| - 2)$ and, by the induction hypothesis, $G[(V(G) \setminus V(B)) \cup \{x\}]$ has at most $(q-1)\binom{b}{2} + 2(|V(G)| - |V(B)| + 1)$ edges. Therefore, $|E(G)| \leq (q-1)\binom{b}{2} + 2(|V(G)| - |V(B)| + 1) + \binom{b}{2} + 2(|V(B)| - 2) = q\binom{b}{2} + 2(|V(G)| - 1)$, which proves the statement. This concludes the proof of (4.2). ∎

Since finding the block decomposition of a graph G can be done in $O(|V(G)| + |E(G)|)$ time (see, e.g., [9]), (4.2) has the following corollary:

(4.3) *Let b be a fixed integer and let G be a* GOLoP(b) *graph. Finding the block decomposition of G can be done in $O(|V(G)|)$ time.*

4.2 K-Match(G, r, k) for GOLoP graphs

In this section, it will be notationally more convenient to think of an edge coloring of a multigraph $H = (G, \text{mp})$ as a schedule. We need to introduce a bit more notation. A *schedule* (of length k) is a function $S : \{1, \ldots, k\} \to \mathcal{M}_H$. For $e \in E(H)$, let $T_S(e) = \sum_{t=1}^{k} \mathbb{1}(e \in S(t))$ (where $\mathbb{1}$ is the indicator function). Informally speaking, $T_S(e)$ is the total amount of time schedule S spends on edge e. Likewise, for vertices $u, v \in V(H)$, let $T_S(u, v) = \sum_{t=1}^{k} \mathbb{1}(S(t) \text{ covers both } u \text{ and } v)$ and, for a matching $M \in \mathcal{M}_H$, let $T_S(M) = \sum_{t=1}^{k} \mathbb{1}(S(t) = M)$. A schedule S is said to be *feasible for* H if $T_S(e) = \text{mp}(e)$ for all $e \in E(H)$. We state the following two observations without a proof:

(4.4) *Let H be a multigraph and let k be an integer. Then, $\chi'(H) \leq k$ if and only if there exists a feasible schedule of length k for H.*

(4.5) *Let $H = (G, \mathrm{mp})$ be a multigraph and let x be a cut-vertex of H. Let $K^1, \ldots, K^p$ be the connected components of $H - x$. Then,*

$$\chi'(H) = \max\left[\deg(x),\ \max_{i=1,\ldots,p}\left\{\chi'(H[V(K^i) \cup \{x\}])\right\}\right].$$

This latter observation allows us to concentrate on the blocks of GOLoP multigraphs. To deal with the sets of clones in GOLoP multigraphs, we start with a lemma for bipartite multigraphs in which one side of the bipartition has exactly two vertices.

(4.6) *Let F be a bipartite multigraph on vertex sets X, Y with $X = \{u, v\}$. Then, for $\tau \in \mathbb{Z}_+$, there exists a feasible schedule S for F such that $T_S(u,v) = \tau$ if and only if*

$$\tau \leq \deg(u) + \deg(v) - \Delta(F). \tag{5}$$

Moreover, for all τ satisfying (5), there exists a feasible schedule of length $\deg(u) + \deg(v) - \tau$ such that $T_S(u,v) = \tau$.

Proof. Since we are always working with F in this proof, we will drop the subscript F from $\mathcal{M}_F$, $\mathcal{M}_F(e)$ and $\mathcal{M}_F(u,v)$. Let

$$T^* = \max_{\mathbf{x} \in \mathbb{Z}_+^{\mathcal{M}}} \left\{ \sum_{M \in \mathcal{M}(u,v)} x(M) \;\middle|\; \sum_{M \in \mathcal{M}(e)} x(M) = \deg(e) \text{ for all } e \in E(F) \right\}. \tag{6}$$

We first claim that:

$(*)$ $\quad T^* = \deg(u) + \deg(v) - \Delta(F)$.

Consider an optimal solution $x(M)$ to the optimization problem in (6). Let $\mathcal{A}_u = \mathcal{M}(u) \setminus \mathcal{M}(v)$ and $\mathcal{A}_v = \mathcal{M}(v) \setminus \mathcal{M}(u)$. Notice that the sets $\mathcal{A}_u$, $\mathcal{A}_v$ and $\mathcal{M}(u,v)$ are disjoint and that $\mathcal{M} = \mathcal{A}_u \cup \mathcal{A}_v \cup \mathcal{M}(u,v) \cup \{\emptyset\}$. We may without loss of generality assume that $x(\emptyset) = 0$. Observe that

$$\sum_{M \in \mathcal{A}_u \cup \mathcal{M}(u,v)} x(M) = \deg(u), \qquad \text{and} \qquad \sum_{M \in \mathcal{A}_v \cup \mathcal{M}(u,v)} x(M) = \deg(v).$$

This implies that

$$\sum_{M \in \mathcal{M}} x(M) = \deg(u) + \deg(v) - \sum_{M \in \mathcal{M}(u,v)} x(M),$$

and, by replacing the objective in (6) and taking out the constants, it follows that

$$T^* = \deg(u) + \deg(v) - \min_{\mathbf{x} \in \mathbb{Z}_+^{\mathcal{M}}} \left\{ \sum_{M \in \mathcal{M}} x(M) \;\middle|\; \sum_{M \in \mathcal{M}(e)} x(M) = \deg(e) \text{ for all } e \in E(F) \right\}. \tag{7}$$

Since the optimization problem in the right hand side of this equation has optimal value $\chi'(F)$ (see (3)) and $\chi'(F) = \Delta(F)$ by Kőnig's theorem [15], this proves $(*)$. $\square$

It follows from (∗) that it suffices to prove that there exists a feasible schedule S for F such that $T_S(u,v) = \tau$ if and only if $\tau \leq T^*$. So assume that there exists a feasible schedule S with $T_S(u,v) = \tau \geq T^* + 1$. Then, setting $x(M) = T_S(M)$ for all $M \in \mathcal{M}$, we obtain a solution for (6) with objective value $\tau \geq T^* + 1$, a contradiction.

Next, suppose that $0 \leq \tau \leq T^*$. Let $x(M)$ be an optimal solution to (6). Let $M_1, \dots, M_q$ be a sequence of matchings constructed by repeating each matching $M \in \mathcal{M}$ exactly $x(M)$ times. Thus, $q = \sum_{M\in\mathcal{M}} x(M) = \Delta(F)$. We can arrange this sequence in such a way that the first T^* matchings are matchings in $\mathcal{M}(u,v)$ and the remaining $q - T^*$ matchings are matchings in $\mathcal{M} \setminus \mathcal{M}(u,v)$. Construct a schedule S as follows:

- For $t = 1, \dots, \tau$, set $S(t) = M_t$.
- For $i = 1, \dots, T^* - \tau$, write $\{e_1, e_2\} = M_{\tau+i}$ and set $S(\tau + 2i - 1) = \{e_1\}$, $S(\tau + 2i) = \{e_2\}$.
- For $i = 1, \dots, q - T^*$, set $S(2T^* - \tau + i) = M_{T^*+i}$.

It is straightforward to check that S is feasible and $T_S(u,v) = \tau$. Moreover, the length of S is $2T^* - \tau + q - T^* = q + T^* - \tau = \deg(u) + \deg(v) - \tau$. This proves (4.6). ■

Having dealt with the sets of clones in GOLoP multigraphs, we can now prove the following:

(4.7) *Let b be a fixed integer and let H be a* GOLoP(b) *multigraph. Then, $\chi'(H)$ can be determined in $O(|V(H)|)$ time.*

Proof. Let $H = (G, \mathrm{mp}_H)$. Since, by (4.3), the block-decomposition of a GOLoP(b) graph can be found in $O(|V(G)|) = O(|V(H)|)$ time, it follows from (4.5) that it suffices to prove the lemma for the blocks of H. So we may assume that H is 2-connected.

Let (G', W) be the collapsed graph associated with G and let $p, C_1, \dots, C_p, F_1, \dots, F_p$ as in the definition of the collapsed graph. By the maximality of the sets $C_1, \dots, C_p$, every set of clones C_i has a unique pair of common neighbors $\{u_i, v_i\}$. Since u_i and v_i are the vertices of the graph on at most b vertices from which H was constructed by iteratively cloning vertices of degree two, there are at most $\binom{b}{2}$ choices of u_i and v_i and, hence, $p \leq \binom{b}{2}$. Let $T_i^* = \deg_{F_i}(u_i) + \deg_{F_i}(v_i) - \Delta(F_i)$.

For conciseness, we will write $\mathcal{M}'$ for $\mathcal{M}_{G'}$ and $\mathcal{M}'(e)$ for $\mathcal{M}_{G'}(e)$. We will construct an integer linear programming problem whose objective value is $\chi'(H)$, and whose variables correspond to the matchings in G'. The idea is that the edges u_iv_i in matchings in G' will play the role of pairs of edges $\{u_ic, v_ic'\}$ in matchings in H with distinct $c, c' \in C_i$. Consider the following integer linear programming problem.

$$z^* = \underset{\mathbf{x}\in\mathbb{Z}_+^{\mathcal{M}'}}{\text{minimize}} \quad \sum_{M\in\mathcal{M}'} x(M) \tag{8}$$

$$\text{s.t.} \quad \sum_{M\in\mathcal{M}'(e)} x(M) = \mathrm{mp}_H(e) \qquad \text{for all } e \in E(G') \setminus \bigcup_{i=1}^{p} \{u_ia_i, v_ia_i, u_iv_i\} \tag{8a}$$

$$\sum_{\substack{M\in\mathcal{M}'(za_i)\\ \cup\mathcal{M}'(u_iv_i)}} x(M) = \sum_{c\in C_i} \mathrm{mp}_H(zc) \qquad \text{for all } z \in \{u_i, v_i\}, i \in [p] \tag{8b}$$

$$\sum_{M\in\mathcal{M}'(u_iv_i)} x(M) \le T_i^* \qquad \text{for all } i\in[p]. \tag{8c}$$

Constructing the problem means calculating the values of T_i^*, which can clearly be done in $O(|C_i|)$ time. Since $p \le \binom{b}{2}$, the problem is an integer linear programming problem with $O(1)$ variables and constraints. Using Eisenbrand's algorithm for integer linear programming in fixed dimension [7], this problem can be solved in $O(1)$ time. Thus, the overall complexity of computing z^* is $O(|V(H)|)$.

We claim that $z^* = \chi'(H)$. First, to prove that $z^* \ge \chi'(H)$, we claim that any solution to (8) can be turned into a feasible schedule for H of length z^* with the help of (4.6). To see this, consider an optimal solution $\{x(M)\}_{M\in\mathcal{M}'}$ of (8). We can, by the constraints of (8), construct a function $S' : \{1,\dots,z^*\} \to \mathcal{M}'$ such that $T_{S'}(e) = \mathrm{mp}_H(e)$ for all $e \in E(G')\setminus\bigcup_{i=1}^{p}\{u_ia_i, v_ia_i, u_iv_i\}$, $T_{S'}(za_i) + T_{S'}(u_iv_i) = \sum_{c\in C_i}\mathrm{mp}_H(zc)$ for all $z\in\{u_i,v_i\}, i\in[p]$, and $T_{S'}(u_iv_i) \le T_i^*$ for all $i\in[p]$. S' is not a schedule because S' is defined on the matchings of G' and not on the matchings of H.

We will turn S' into a schedule for H as follows. Let $S^{(0)} = S'$. We will iteratively construct a sequence $S^{(1)},\dots,S^{(p)}$ of functions from $\{1,\dots,z^*\}$ to $\mathcal{M}'\cup\mathcal{M}_H$, the last of which will be a schedule for H. For $i\in[p]$, do the following. Let $\tau_i = \sum_{M\in\mathcal{M}'(u_iv_i)} x(M)$. By (8c), $\tau_i \le T_i^*$. It follows from (4.6) that there exists a feasible schedule $S_i : \{1,\dots,\deg_{F_i}(u_i)+\deg_{F_i}(v_i)-\tau_i\} \to \mathcal{M}_{F_i}$ such that $T_{S_i}(u_i,v_i) = \tau_i$. Let $Z_i = \{t \mid S^{(i-1)}(t) \in \mathcal{M}'(u_ia_i)\cup\mathcal{M}'(v_ia_i)\cup\mathcal{M}'(u_iv_i)\}$. It follows from (8b) that

$$|Z_i| = \sum_{c\in C_i}\mathrm{mp}_H(u_ic) + \sum_{c\in C_i}\mathrm{mp}_H(v_ic) - \tau_i = \deg_{F_i}(u_i) + \deg_{F_i}(v_i) - \tau_i.$$

Let $\phi_i : Z_i \to \{1, \deg_{F_i}(u_i) + \deg_{F_i}(v_i) - \tau_i\}$ be a bijection such that, for all $t\in Z_i$, $S_i(\phi_i(t))$ covers both u_i and v_i if and only if $u_iv_i \in S^{(i-1)}(t)$. Now, for all $t\in\{1,\dots,z^*\}$, set

$$S^{(i)}(t) = \begin{cases} \left(S^{(i-1)}(t)\setminus\{u_ia_i, v_ia_i, u_iv_i\}\right)\cup S_i(\phi_i(t)) & \text{for } t\in Z_i,\\ S^{(i-1)}(t) & \text{otherwise.}\end{cases}$$

Observe that $T_{S^{(i)}}(u_ic) = \mathrm{mp}_{F_i}(u_ic)$ and $T_{S^{(i)}}(v_ic) = \mathrm{mp}_{F_i}(v_ic)$ for all $c\in C_i$. Moreover, $T_{S^{(i)}}(e) = T_{S^{(i-1)}}(e)$ for all $e\in E(H)\cup E(G')\setminus(E(F_i)\cup\{u_ia_i, v_ia_i, u_iv_i\})$.

After having done this for all $i\in[p]$, let $S = S^{(p)}$. Then, $S(t)\in\mathcal{M}_H$ for all $t\in\{1,\dots,z^*\}$ and $T_S(e) = \mathrm{mp}_H(e)$ for all $e\in E(H)$. Thus, S is a feasible schedule of length z^* for H, implying by (4.4) that $\chi'(H)\le z^*$.

To prove that $\chi'(H)\ge z^*$, consider a schedule $S : [\chi'(H)]\to\mathcal{M}_H$ of length $\chi'(H)$. Such a schedule exists because of (4.4). For all $t\in[\chi'(H)]$, let $I_u(t)\subseteq[p]$ (resp. $I_v(t)$) be the set of all indices i such that $u_ic\in S(t)$ (resp. $v_ic\in S(t)$) for some $c\in C_i$. Define

$$S'(t) = \left(S(t)\setminus\bigcup_{i=1}^{p}E(F_i)\right)\cup\left(\bigcup_{i\in I_u(t)\setminus I_v(t)}\{u_ia_i\}\right)\cup\left(\bigcup_{i\in I_v(t)\setminus I_u(t)}\{v_ia_i\}\right)\cup\left(\bigcup_{i\in I_u(t)\cap I_v(t)}\{u_iv_i\}\right)$$

and, for $M \in \mathcal{M}'$, let $x(M) = T_{S'}(M)$. We claim that x is a solution of (8). It is straightforward to see that $S'(t) \in \mathcal{M}'$ for all t. Next, observe that, for all $e \in E(G') \setminus \cup_{i=1}^{p}\{u_i a_i, v_i a_i, u_i v_i\}$,

$$\sum_{M \in \mathcal{M}'(e)} x(M) = T_{S'}(e) = \left|\{t : e \in S'(t)\}\right| = |\{t : e \in S(t)\}| = T_S(e) = \mathrm{mp}_H(e).$$

Moreover, we have, for each za_i with $z \in \{u_i, v_i\}$ and $i \in [p]$,

$$\sum_{\substack{M \in \mathcal{M}'(za_i) \\ \cup \mathcal{M}'(u_i v_i)}} x(M) = T_{S'}(za_i) + T_{S'}(u_i v_i) = |\{t : zc \in S(t), c \in C_i\}| = \sum_{c \in C_i} T_S(zc) = \sum_{c \in C_i} \mathrm{mp}_H(zc).$$

Next, observe that the schedule S implies a schedule S_i for F_i. It follows from (4.6) that $T_{S_i}(u_i, v_i) \leq T_i^*$. Therefore,

$$\sum_{M \in \mathcal{M}'(u_i, v_i)} x(M) = T_{S'}(u_i, v_i) = |\{t : i \in I_u(t) \cap I_v(t)\}| = T_{S_i}(u_i, v_i) \leq T_i^*.$$

This proves that $x(M)$ is a solution to (8), thereby proving that $z^* \leq \chi'(H)$. This proves (4.7). ■

This resolves our second problem.

(4.8) *Let b and k be fixed integers. Let G be a* GOLoP(b) *graph, let r be a rate function for G. Then* K-MATCH(G, r, k) *can be solved in $O(|V(G)|)$ time.*

Proof. It follows from (2.1) that the answer to K-MATCH(G, r, k) is YES if and only if $\chi'(G, \lceil kr \rceil) \leq k$. The theorem follows from (4.7). ■

4.3 MATCH(G, r) for GOLoP graphs

In this section, we focus on MATCH(G, r) for GOLoP graphs. We will use an algorithm that is very similar to the algorithm used for the K-MATCH(G, r, k) problem in the previous section. We will do this by proving continuous versions of the results from the previous section.

By duplicating edges, (4.5) generalizes easily to the setting of the fractional chromatic index of weighted graphs. To be precise, we have the following:

(4.9) *Let F be a graph, let r be a rate function for $E(F)$ and let $K^1, \ldots, K^p$ be the connected components of $F - x$. Then, letting $r_i = r|_{E(F|K_i)}$ for $i \in [p]$, it holds that*

$$\chi'_f(F, r) = \max\left[\sum_{e \in \delta(x)} r(e),\ \max_{i=1,\ldots,p}\left\{\chi'_f(F[V(K^i) \cup \{x\}], r_i)\right\}\right].$$

As in the previous section, it will be notationally more convenient to think of a fractional edge coloring of a weighted graph as a schedule. Other than in the previous section, our schedule will now be a function that is defined on a continuous time range. Let us make some definitions. For

$T \geq 0$, a *schedule* (of length T) is a piecewise constant function $S : [0, T] \to \mathcal{M}_G$. For $e \in E(G)$, let $T_S(e) = \int_0^T \mathbb{1}(e \in S(t))\mathrm{d}t$ (where $\mathbb{1}$ is the indicator function). Informally speaking, $T_S(e)$ is the total amount of time schedule S spends on edge e. Likewise, for vertices $u, v \in V(G)$, let $T_S(u, v) = \int_0^T \mathbb{1}(S(t)$ covers both u and $v)\mathrm{d}t$ and, for a matching $M \in \mathcal{M}_G$, let $T_S(M) = \int_0^T \mathbb{1}(S(t) = M)\mathrm{d}t$. A schedule S is said to be *r-feasible (for G)* if $T_S(e) = r(e)$ for all $e \in E(G)$.

We state the following obvious result without a proof:

(4.10) *$\chi'_f(G, r) \leq t$ if and only if there exists an r-feasible schedule of length t for G.*

We have the following fractional version of Kőnig's theorem, which easily follows from Kőnig's original theorem by standard compactness arguments.

(4.11) *Let G be a bipartite graph and let r be a rate function for G. Then $\chi'_f(G, r) = \max_{u \in V(G)}\{r(u)\}$.*

We start with the following continuous version of (4.6).

(4.12) *Let F be a bipartite graph on vertex sets X, Y with $X = \{u, v\}$ and let r be a rate function for H. Then there exists an r-feasible schedule S for F such that $T_S(u, v) = \tau$ if and only if $0 \leq \tau \leq r(u) + r(v) - \chi'_f(F, r)$. Moreover, if $0 \leq \tau \leq r(u) + r(v) - \chi'_f(F, r)$, then there exists an r-feasible schedule of length $r(u) + r(v) - \tau$ such that $T_S(u, v) = \tau$.*

Proof. First suppose that $0 \leq \tau \leq r(u) + r(v) - \chi'_f(F, r)$. Since $r(e)$ is rational for all $e \in E(F)$, there exists a positive integer d such that $dr(e)$ is integral for all $e \in E(F)$. Notice that $\chi'_f(F, r) = \frac{1}{d}\chi'(F, rd) = \frac{1}{d}\Delta(F, rd))$ and $r(e) = \frac{1}{d}\mathrm{mp}_{(F,rd)}(e)$. Since $d\tau \leq dr(u) + dr(v) - d\chi'_f(F, r) = \mathrm{mp}_{(F,rd)}(u) + \mathrm{mp}_{(F,rd)}(v) - \Delta(F, rd)$, it follows from (4.6) that there exists a feasible schedule S' for (F, rd) such that $T_{S'}(u, v) = d\tau$ and such that S' is of length $dr(u) + dr(v) - d\chi'_f(F, r)$. Clearly, S' yields an r-feasible schedule S for F such that $T_S(u, v) = \tau$ and the length of S is $r(u) + r(v) - \tau$.

Conversely, suppose that $\tau > r(u) + r(v) - \chi'_f(F, r)$ and suppose that there exist r-feasible schedule S for F such that $T_S(u, v) = \tau$. Since such a schedule can be found by means of a linear program with rational parameters, we may assume that $T_S(e)$ is rational for all $e \in E(G)$. In particular, there exists an integer d such that $dT_S(M)$ and $dT_S(e)$ is integral for all $M \in \mathcal{M}_F$ and all $e \in E(F)$. Now construct a function $S' \colon \{1, \ldots, d|S|\} \to \mathcal{M}_F$ by repeating matching M exactly $dT_S(M)$ times. Clearly, S' is a feasible schedule for $M(F)^{r,d}$ such that $T_{S'}(u, v) = dT_S(u, v) = d\tau$ while $d\tau > dr(u) + dr(v) - d\chi'_f(F, r) = \mathrm{mp}_{(F,rd)}(u) + \mathrm{mp}_{(F,rd)}(v) - \Delta(F, rd)$, contrary to (4.6). ■

The previous claim allows us to deal with the blocks of GOLoP graphs:

(4.13) *Let G be a graph in* GOLoP(b) *and let r be a rate function for G. Then, $\chi'_f(G, r)$ can be determined in $O(|V(G)|)$ time.*

Proof. As in the proof of (4.13), we may assume that G is 2-connected because, by (4.3), the block-decomposition of a GOLoP(b) graph can be found in $O(|V(G)|)$ time. Let p, $C_1, \ldots, C_p$, u_i, v_i, G', $a_1, \ldots, a_p$, $\mathcal{M}'$, $\mathcal{M}'(e)$ be as in (4.7). (See Figure 1.) Let $H_i = G[C_i \cup \{u_i, v_i\}]$, $r_i = r|_{E(H_i)}$, and $T_i^{**} = r_i(u_i) + r_i(v_i) - \chi'_f(H_i, r_i)$. Consider the following linear program.

$$
\begin{aligned}
z^{**} = \quad & \underset{\mathbf{x} \in \mathbb{R}_+^{\mathcal{M}'}}{\text{minimize}} & & \sum_{M \in \mathcal{M}'} x(M) & & \\
& \text{subject to} & & \sum_{M \in \mathcal{M}'(e)} x(M) = r(e) & & \text{for all } e \in E(G') \setminus \bigcup_{i=1}^{p} \{u_i a_i, v_i a_i, u_i v_i\} \qquad (9) \\
& & & \sum_{\substack{M \in \mathcal{M}'(z a_i) \\ \cup \mathcal{M}'(u_i v_i)}} x(M) = \sum_{c \in C_i} r(zc) & & \text{for all } z \in \{u_i, v_i\}, i \in [p] \\
& & & \sum_{M \in \mathcal{M}'(u_i v_i)} x(M) \le T_i^{**} & & \text{for all } i \in [p].
\end{aligned}
$$

Since $p \le \binom{b}{2}$, it follows that this linear program contains a constant number of variables and constraints. Constructing the linear program means calculating the values of T_i^{**}, which can be done in $O(|C_i|)$ time by (4.11). Thus, the overall complexity of computing z^{**} is $O(|V(G)|)$.

We claim that $z^{**} = \chi'_f(G, r)$. To see this, let $x(M)$ be an optimal solution of (9). Since the right-hand side values of the constraints in (9) are rational, we may assume that $x(M)$ is rational for all $M \in \mathcal{M}'$. Therefore, there exists an integer d such that $dx(M)$ and $dr(e)$ are integral for all $M \in \mathcal{M}'$ and all $e \in E(G)$. Now, observe that (9) is the LP-relaxation of (8) in (4.7) applied to the multigraph $(G, \lceil dr \rceil)$, where all the right-hand side values of the constraints have been multiplied by d. In particular, notice that the value of T_i^* in (8) satisfies

$$T_i^* = \mathrm{mp}_{(H_i, rd)}(u) + \mathrm{mp}_{(H_i, rd)}(v) - \Delta(H_i, rd) = dT_i^{**}.$$

Thus, $dx(M)$ is an optimal solution of (8). Let z^* be the corresponding optimal objective value. It follows from the proof of (4.7) that $dz^{**} = z^* = \chi'(G, dr) = d\chi'_f(G, r)$, which implies that $z^{**} = \chi'_f(G, r)$. This proves (4.13). ∎

It now follows immediately from (2.2) and (4.13) that $\textsc{Match}(G, r)$ can be solved in linear time for GOLoP graphs:

(4.14) *Let $b \ge 1$. Let G be a* GOLoP(b) *graph and let r be a rate function for G. The problem* $\textsc{Match}(G, r)$ *can be solved in $O(|V(G)|)$ time.*

4.4 $\textsc{Find-Match}(G, r, k)$ for GOLoP graphs

We have described, in Section 4.2, a linear-time algorithm to verify whether the chromatic index of a GOLoP multigraph $(G, \lceil kr \rceil)$ is at most k. In this section, we show that this algorithm can be turned into a quadratic-time algorithm to find a schedule of length k for G with respect to r, if such a schedule exists. In order to make the computation efficient, we store each schedule as a set of pairs (M, ℓ), where M is a matching in G and ℓ is the number of occurences of M in the schedule.

We start by obtaining a pseudo-schedule $\mathcal{Z}$ being a collection of schedules for all blocks of G, and then show how to assemble the schedule for G using $\mathcal{Z}$. We begin with details of $\mathcal{Z}$. Let G be a graph, let $B^1, \dots, B^q$ be the blocks of G, and let $C_1^j, \dots, C_{p_j}^j$ be the sets of nonadjacent clones in block B^j. For $j \in [q]$ and $i \in [p_j]$, let $(B^{j\prime}, C_i^j)$ be the collapsed graph associated with B^j, and let u_i^j, v_i^j be the common neighbors of the vertices in C_i^j. Finally, let $F_i^j = G[C_i^j \cup \{u_i^j, v_i^j\}]$. Define a *pseudo-schedule* for G as $\mathcal{Z} = \big((Z^1, Z_1^1, \dots, Z_{p_1}^1), \dots, (Z^q, Z_1^q, \dots, Z_{p_q}^q)\big)$, where Z^j is a schedule for $B^{j\prime}$ and Z_i^j is a schedule for F_i^j. Note that, each Z^j and Z_i^j is a list of pairs (M, ℓ), with M being a matching in the relevant graph and ℓ being the number of occurences of M in the corresponding schedule. Define

$$
\begin{aligned}
c(\mathcal{Z}, x) &= \sum_{j \in [q]} \sum_{\substack{(M,\ell) \in Z^j \\ \text{and } M \text{ covers } x}} \ell, && \text{for any cut-vertex } x \text{ of } G, \\
|Z^j| &= \sum_{(M,\ell) \in Z^j} \ell, && \text{for } j \in [q], \\
|\mathcal{Z}| &= \max\left\{ \max_{j \in [q]} |Z^j|, \quad \max\{c(\mathcal{Z}, x) : x \text{ is a cut-vertex of } G\} \right\}.
\end{aligned}
$$

Here, $|Z^j|$ denotes the length of schedule Z^j, $c(\mathcal{Z}, x)$ denotes the number of matchings in $\mathcal{Z}$ that cover x, and $|\mathcal{Z}|$ denotes the length of pseudo-schedule $\mathcal{Z}$.

First we give an algorithm that computes $\mathcal{Z}$ of length k in linear time.

Algorithm FIND-PSEUDO-SCHEDULE

Input: A GOLoP graph G and a rate function r for $E(G)$.
Output: Either a pseudo-schedule $\mathcal{Z}$ of length at most k for G with respect to r, or the message that no such pseudo-schedule exists.

(1) Find the block decomposition $B^1, \dots, B^q$ of G.

(2) If any cut-vertex $x \in V(G)$ satisfies $\sum_{e \in \delta(x)} \lceil r(e)k \rceil > k$, then terminate because no schedule of length at most k exists for G with respect to r.

(3) For each $j \in [q]$:

  (3a) Construct $H^j = (B^j, \lceil kr_j \rceil)$, where $r_j = r|_{E(B_j)}$.

  (3b) Identify the sets $C_1^j, \dots, C_{p_j}^j$ and the vertices u_i^j and v_i^j, $i \in [p_j]$.

  (3c) Solve the integer linear program (8) corresponding to $B^{j\prime}$ to construct Z^j. If the optimal value of the IP is greater than k, terminate because no schedule of length at most k exists for G with respect to r.

  (3d) For $i \in [p_j]$, construct a schedule Z_i^j of length $\deg_{F_i^j}(u_i^j) + \deg_{F_i^j}(v_i^j) - \tau_i^j$ for F_i^j such that $T_{Z_i^j}(u_i^j, v_i^j) = \tau_i^j$.

Before we prove the correctness of this algorithm, we need a small technical lemma that allows us to easily construct the sets $C_1, \dots, C_p$.

(4.15) *Let G be a graph, let $c \in V(G)$ be such that $\deg_G(c) = 2$ and let u, v be the neighbors of c. Let G' be obtained from G by non-adjacent cloning of c. If G' has no cut-vertex, then either*

$\min\{\deg_{G'}(u), \deg_{G'}(v)\} \geq 3$*, or* G' *is a 4-cycle.*

Proof. Let $c_1 \neq c$ be a clone of c. If $V(G') = \{u, v, c, c_1\}$, then G' is a 4-cycle and the claim holds. So we may assume that there exists $x \in V(G') \setminus \{u, v, c, c_1\}$. Because G' is 2-connected, it follows that there exist paths P_1, P_2 from x to c_1 in G' such that $V(P_1) \cap V(P_2) = \{x, c_1\}$. Since $N_{G'}(c_1) = \{u, v\}$, it follows that one of P_1, P_2 contains u and not v, and the other contains v and not u. This implies that $\deg_{G'}(u) \geq 3$ and $\deg_{G'}(v) \geq 3$. ■

This allows us to prove the correctness and the complexity of algorithm Find-Pseudo-Schedule.

(4.16) *Let* $b \geq 1$*, let* G *be a* GOLoP(b) *graph and let* r *be a rate function for* G*. Then, algorithm* Find-Pseudo-Schedule *either finds a pseudo-schedule of length at most* k *for* G *with respect to* r*, or determines that no such pseudo-schedule exists. Its running time is* $O(|V(G)|)$*.*

Proof. It follows from (4.3) that steps (1) and (2) can be done in $O(|V(G)|)$ time. Now, let B^j a block of G. Let $H_j = (B^j, \lceil r_j k \rceil)$. Clearly, H_j can be constructed in $O(|V(B^j)|)$ time because $|E(B^j)| = O(|V(B^j)|)$, as proved in [2].

Identifying the sets $C_1^j, \ldots, C_{p_j}^j$ of non-adjacent clones can be done as follows. If B^j is a 4-cycle, then $p = 1$ and we may take C_1^j to be any two non-adjacent vertices in $V(B^j)$. Otherwise, it follows from (4.15) that $\deg(u_i^j), \deg(v_i^j) \geq 3$ for all $i \in [p]$. By the definition of GOLoP(b), there are at most b such vertices and at most $\binom{b}{2} = O(1)$ pairs of such vertices. Thus, we can identify the sets C_i^j by constructing the list of all pairs $\{u, v\}$ of vertices u, v of degree at least three and assign every vertex of degree two to exactly one of these pairs. Let $C(u, v)$ be the vertices of degree two that were assigned to the pair $\{u, v\}$. Then, $\{C_1^j, \ldots, C_{p_j}^j\} = \{C(u, v) : |C(u, v)| \geq 2\}$. Thus, step (3b) takes $O(|V(B^j)|)$ time.

To construct a pseudo-schedule of length at most k for H, we first need to solve the corresponding integer program (8). As shown in the proof of (4.7), this can be done in $O(|V(B^j)|)$ time. Constructing Z^j from the solution can be trivially done in $O(1)$ time since $|\mathcal{M}_{B^{j\prime}}| = O(1)$.

Next, we need to solve the integer program in (7). To do so, we view this problem as an open shop scheduling problem with two machines, in which u_i^j and v_i^j are the two machines, C_i^j is the set of jobs, and job $c \in C_i^j$ needs to be processed on machine u_i^j for $\mathrm{mp}(u_i^j c)$ time units and on machine v_i^j for $\mathrm{mp}(v_i^j c)$ time units. Minimizing the makespan of this scheduling instance is equivalent to finding a minimum length schedule for F_i^j. It was shown in [8] that finding an optimal solution for this open shop scheduling can be done in linear time, i.e. in $O(|V(C_i^j)|)$ time. We note here that preemptions do not result in shorter optimal schedules for the two-machine open shop makespan problem. Therefore, optimal non-preemptive schedules can be taken intead of optimal preemptive ones to find a minimum length schedule for F_i^j and thus an optimal solution to (7). The obvious advantage of this choice is the reduction in the number of distinct matchings required by the optimal solution, this number is obviously $O(|V(C_i^j)|)$. Thus, modifying the solution to obtain a schedule Z_i^j such that $T_{Z_i^j}(u_i^j, v_i^j) = \tau_i^j$ and that has length $\deg_{F_i^j}(u_i^j) + \deg_{F_i^j}(v_i^j) - \tau^j$ takes again $O(|C_i^j|)$ time. This proves (4.16). ■

Now we prove that, given a pseudo-schedule $\mathcal{Z}$ of length at most k, we can assemble a sequence of at most k matchings in G. Furthermore, this sequence requires at most $O(|V(G)|)$ distinct matchings in G, which permits a succinct encoding of schedule of G that specifies a list of pairs (M, ℓ), with M being a matching in the G and ℓ being the number of occurences of M in the schedule.

We begin by giving an informal description of the method. Consider a block-cutpoint tree T of G rooted at any vertex. We start by ordering the blocks of G, which is equivalent to ordering the vertices of T, so that for each $i \geq 1$ the first i blocks of the order induce a connected subgraph of G. This can be achieved by, for example, taking any depth-first-search ordering of the vertices in T. Recall that $\mathcal{Z}$ gives a feasible schedule, or equivalently a sequence of matchings, for each block of G. Each matching of G is then assembled from the block matchings as follows. We start with an empty matching M of G and then process the blocks according to their previously fixed order to construct M iteratively. If the cut-vertex connecting the current block with its 'parent' block is an endvertex of an edge already in M, then we try to find a matching in the current block not saturating the cut-vertex. If such a matching exists, then its edges are added to M. If there is no edge in M that saturates the cut-vertex, then we take a matching, if any, in the current block that saturates the cut-vertex and we add its edges to M. The matchings in the blocks are obtained from the corresponding schedules for the blocks. When all blocks are processed, then we add the matching M thus assembled to the output sequence of matchings, and simultaneously update the block schedules corresponding to the block matchings used by M. We then proceed to the next iteration to find the next matching of G. In (4.17) we prove that this method gives the desired sequence of k matchings for G.

We are now ready to give a more formal pseudo-code of an algorithm that constructs the desired sequence of matchings from the pseudo-schedule $\mathcal{Z}$.

Algorithm Recover-Schedule

Input: A GOLoP graph G, the block-decomposition of G, and a pseudo-schedule $\mathcal{Z}$ of length $k \leq |\mathcal{Z}|$.

Output: A schedule S for G given as a set $\{(M, \ell) \colon M \in \mathcal{M}_G, \ell \in \mathbb{Z}_+\}$.

(1) Order the blocks $B^1, \dots, B^q$ so that $G\big[\bigcup_{l=1}^{j-1} V(B^l)\big]$ is connected for all $j \in [q]$. Set $S := \emptyset$.

(2) While there exists $j \in [q]$ such that $Z^j \neq \emptyset$ do the following:

- (2a) Set $M := \emptyset$.
- (2b) For each $j \in [q]$ with $Z^j \neq \emptyset$, do the following:
  - If $j = 1$, choose M^1 such that $(M^1, \ell) \in Z^1$ and set $\ell' := \ell$.
  - If $j > 1$, then let x^j be the unique cut-vertex of $G\big[\bigcup_{l=1}^{j-1} V(B^l)\big]$ that lies in $V(B^j)$. If M saturates x^j, then choose any M^j such that $(M^j, \ell) \in Z^j$ does not saturate x^j, if any. If M does not saturate x^j, then choose M^j that saturates x^j, if any. If no M^j saturating x^j exists, then take any M^j. If no such M^j exists, go to step (2b) and consider the next value of j, otherwise set $\ell' := \min\{\ell', \ell\}$.
  - For all $i \in [p_j]$ such that $M^j \cap \{u_i^j v_i^j, u_i^j a_i^j, v_i^j a_i^j\} \neq \emptyset$, do the following:
    - If $u_i^j v_i^j \in M^j$, then choose $(M_i^j, \ell) \in Z_i^j$ such that M_i^j covers both u_i^j and v_i^j, and set $\ell' := \min\{\ell', \ell\}$.
    - If $u_i^j a_i \in M^j$, then choose $(M_i^j, \ell) \in Z_i^j$ such that M_i^j covers u_i^j but not v_i^j, and set

$\ell' := \min\{\ell', \ell\}$.

- If $v_i^j a_i \in M^j$, then choose $(M_i^j, \ell) \in Z_i^j$ such that M_i^j covers v_i^j but not u_i^j, and set $\ell' := \min\{\ell', \ell\}$.
- Set $M^j := \Big(M^j \setminus \{u_i^j v_i^j, u_i^j a_i^j, v_i^j a_i^j\}\Big) \cup M_i^j$.

- Let $M := M \cup M^j$.

(2c) For each M^j chosen in step (2b) replace (M^j, ℓ) by $(M^j, \ell-\ell')$ in Z^j and if $\ell-\ell' = 0$, then delete $(M^j, \ell-\ell')$ from Z^j. Similarly, for each M_i^j chosen in step (2b) replace (M_i^j, ℓ) by $(M_i^j, \ell-\ell')$ in Z_i^j, and delete $(M_i^j, \ell'-\ell)$ from Z_i^j if $\ell - \ell' = 0$. Let $S := S \cup \{(M, \ell')\}$.

Notice that since the number of distinct matchings in Z^j is constant and the number of matchings Z_i^j is $O(|V(F_i^j)|)$ it follows that each iteration of the main loop can be done in $O(|V(G)|)$ time. Thus, by (4.16) and by the fact that each iteration 'eliminates' at least one matching in a block of G, the overall complexity of algorithm Recover-Schedule is $O(|V(G)|^2)$.

(4.17) *Let $b \geq 1$ and let G be a GOLoP(b) with a rate function r. There exists a $O(|V(G)|^2)$-time algorithm that finds a schedule S for G. Moreover, the number of pairwise different matchings in S is $O(|V(G)|)$.*

Proof. The schedule S is a result of the execution of Find-Pseudo-Schedule for G and r (which produces a pseudo-schedule $\mathcal{Z}$) and the execution of Recover-Schedule for $\mathcal{Z}$, G and the block decomposition.

Now, observe that it follows from the ordering of the blocks $B^1, \ldots, B^q$ that for each $j > 1$, there exists a unique cut-vertex x^j of $\bigcup_{l=1}^{j-1} V(B^l)$ that lies in $V(B^j)$. To prove the correctness of this algorithm, it suffices to show that after each iteration we have $|\mathcal{Z}| \leq k - t$, where $t \in [k]$ is the sum of the multiplicities of the matchings added to S prior to and in this iteration. We prove this by induction on the number of iterations. The statement is clearly true at the beginning of the first iteration. Now let $\bar{\mathcal{Z}}$ be the pseudo-schedule and let S be the schedule at the beginning of an iteration. Furthermore, let $t = \sum_{(M,\ell)\in S} \ell)$ and let $\mathcal{Z}$ be the pseudo- schedule at the end of the iteration. We denote by ℓ' the multiplicity of the matching choosen in the iteration.

First, suppose for a contradiction that $|Z^j| \geq k - t + 1$ for some $j \in [q]$. Since by induction $|\bar{Z}^j| \leq k - t + \ell'$, it follows that $|Z^j| = |\bar{Z}^j|$. Hence no matching M^j was chosen in this iteration. Thus, $j > 1$. This means that for some $j' < j$, $M^{j'}$ saturates x^j, and every matching in $\bar{Z}^j$ saturates x^j, because $\bar{Z}^j \neq \emptyset$. But this implies that $c(\bar{\mathcal{Z}}, x^j) \geq \ell' + |\bar{Z}^j| > k - t + \ell'$, contrary to the inductive hypothesis. Thus, $|Z^j| \leq k - t$ for all $j \in [q]$.

Second, suppose for a contradiction that $c(\mathcal{Z}, x) \geq k - t + 1$ for some cut-vertex x of G. By induction, we have $c(\bar{\mathcal{Z}}, x) \leq k - t + \ell'$. Therefore, $c(\mathcal{Z}, x) = c(\bar{\mathcal{Z}}, x)$. This implies that none of the matchings M^j chosen in the iteration saturates x. Therefore, all matchings M that saturate x are already included in S and hence $c(\mathcal{Z}, x) = 0$. This however contradics $c(\bar{\mathcal{Z}}, x) \geq k - t + 1 > 0$. Thus, $c(\bar{\mathcal{Z}}, x) \leq k - t$ for every cut-vertex x of G. This proves that $|\mathcal{Z}| \leq k - t$. This completes the proof of (4.17). ■

5 An upper bound on the schedule length

In this section we are interested in finding, for a given graph G and rates r for which K-MATCH(G, r, k) is affirmative for some k, an upper bound for the length of a shortest schedule. To get a handle on this bound, we will focus on the following property. We say that a graph G has the *lcd-property (with constant C)* if for every rate function r, it holds that if K-MATCH(G, r, k) has an affirmative answer for some k, then K-MATCH(G, r, Cd) also has an affirmative answer, where d is the least common denominator for the rate function r. Notice that, in this definition, the value of C only depends on the graph G and not on the rate function r. Notice also that a graph having the lcd-property with constant C does not necessarily have the lcd-property with constant $C+1$. We do, however, have the following property.

Property 5.1. *Let G be a graph having the lcd-property with constant C. Then G also has the lcd-property with constant tC for any positive integer t.*

We first show that the lcd-property is equivalent to a property that relates the fractional chromatic index and the chromatic index of multigraphs associated with G.

(5.2) *Let G be a graph and let $C \geq 1$ be an integer. The following two statements are equivalent:*

(i) *G has the lcd-property with constant C;*

(ii) $\lceil \chi'_f(G, \mathrm{mp}) \rceil = \lceil \frac{1}{C}\chi'(G, C \cdot \mathrm{mp}) \rceil$ *for every function* mp.

Proof. (i) $\implies$ (ii): Let (G, mp) be given. Since $\chi'(H) \geq \chi'_f(H)$ for every multigraph H, we have

$$\left\lceil \frac{1}{C}\chi'(G, C \cdot \mathrm{mp}) \right\rceil \geq \left\lceil \frac{1}{C}\chi'_f(G, C \cdot \mathrm{mp}) \right\rceil = \left\lceil \chi'_f(G, \mathrm{mp}) \right\rceil.$$

To prove the inequality in the other direction, let $p = \lceil \chi'_f(G, \mathrm{mp}) \rceil$. Set $r = \mathrm{mp}/p$ and $d = p/\gcd(p, \mathrm{mp})$. Then, d is the least common denominator of r and we may write $p = td$ for some integer $t \geq 1$. We have

$$\chi'_f(G, r) = \chi'_f\left(G, \frac{\mathrm{mp}}{p}\right) = \frac{\chi'_f\left(G, \mathrm{mp}\right)}{p} \leq 1.$$

Thus, by (2.2), it follows that K-MATCH(G, r, k) has an affirmative answer for some k. By statement (i), K-MATCH(G, r, Cd) has an affirmative answer. This implies that

$$\frac{1}{C}\chi'(G, C \cdot \mathrm{mp}) \leq \frac{t}{C}\chi'\left(G, \frac{C \cdot \mathrm{mp}}{t}\right) = \frac{t}{C}\chi'(G, Cdr) \leq \frac{tCd}{C} = p.$$

Since p is an integer, it follows that in fact $\lceil \frac{1}{C}\chi'(G, C \cdot \mathrm{mp}) \rceil \leq p$. This proves that (ii) holds.

(ii) $\implies$ (i): Let r be such that K-MATCH(G, r, k) has an affirmative answer for some k. It follows from (2.2) that $\chi'_f(G, r) \leq 1$. Let d be the least common denominator for r. It follows that $\chi'_f(G, d \cdot r) \leq d$. Therefore, by (ii),

$$\left\lceil \frac{1}{C}\chi'(G, Cd \cdot r) \right\rceil = \left\lceil \chi'_f(G, d \cdot r) \right\rceil \leq d,$$

which implies that $\frac{1}{C}\chi'(G, Cd \cdot r) \leq d$, as required. ■

In Section 5.1, we will prove that every OLoP graph has the lcd-property with constant $C = 1$. However, the following example shows that not every graph has the lcd-property with constant 1.

Example. *Let G be the Petersen graph (see Figure 2(a)) and let $r(e) = \frac{1}{3}$ for each $e \in E(G)$. The answer to* K-MATCH(G, r, k) *is* NO *for each $k \leq 3$, because $(G, \lceil \frac{k}{3} \rceil)$ is isomorphic to G and*

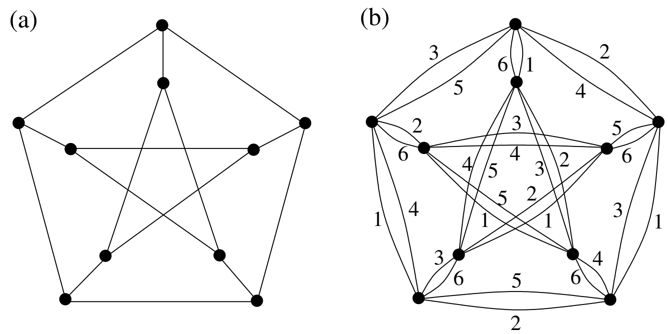

Figure 2: (a) the Petersen graph G; (b) an edge 6-coloring of G_2

$\chi'(G) = 4$. Note that for each $k = 4, 5, 6$ the multigraph $(G, \lceil \frac{k}{3} \rceil)$ is isomorphic to G_2, i.e., the graph obtained by replacing each edge in G by two parallel edges. It is enough to argue that $\chi'(G_2) \geq 6$. This, however, follows from the fact that each matching in the Petersen graph consists of at most 5 edges, and consequently, $E(G_2)$ can not be partitioned into at most 5 matchings, each of size at most 5, because $|E(G_2)| = 30$. Thus, the answer to K-MATCH(G, r, k) *is* NO *for each $k \leq 5$.*

Figure 2(b) gives a 6-edge-coloring of G_2, which implies that the answer to K-MATCH$(G, r, 2d)$ *is* YES*, where $d = 3$ is the least common denominator of the rates. This shows that for the Petersen graph, $C \geq 2$.*

With the Petersen graph in mind, the following question arises: is it true that every graph has the lcd-property for some, graph but not rate function dependent, constant C? And if so, what is the smallest value of C? In Section 5.2, we will prove that the former question always has an affirmative answer, and that, for fixed b, every GOLoP(b) graph satisfies the lcd-property with some constant that depends only on b. Finally, with (5.2) in mind, we point out that the following conjecture of Seymour implies that every graph has the lcd-property with constant $C \leq 2$.

Conjecture 5.3. (Seymour [21]) *For every multigraph H it holds $\lceil \chi'_f(H) \rceil = \lceil \frac{1}{2} \chi'(H_2) \rceil$, where H_2 is the multigraph obtained from H by replacing each edge with two parallel edges.*

This conjecture follows easily from the work of Plantholt and Tipnis [20] for graphs on at most 10 vertices which we now show.

Theorem 5.4. *Let $H = (G, \mathrm{mp})$ be a multigraph such that $|V(G)| \leq 10$. Then, $\left\lceil \frac{1}{2} \chi'(G, 2\mathrm{mp}) \right\rceil = \left\lceil \chi'_f(G, \mathrm{mp}) \right\rceil$.*

Proof. We first claim that is suffices to prove that $\chi'(H) = \left\lceil \chi'_f(H) \right\rceil$ for any $H = (G, \mathrm{mp})$ such that $|V(G)| \leq 10$ and $\mathrm{mp}(e)$ is even for all $e \in E(G)$. Indeed, if $\chi'(G, 2\mathrm{mp}) = \left\lceil \chi'_f(G, 2\mathrm{mp}) \right\rceil$, then

$$\left\lceil \frac{1}{2}\chi'(G, 2\mathrm{mp}) \right\rceil = \left\lceil \frac{1}{2} \left\lceil \chi'_f(G, 2\mathrm{mp}) \right\rceil \right\rceil = \left\lceil \frac{1}{2}\chi'_f(G, 2\mathrm{mp}) \right\rceil = \left\lceil \chi'_f(G, \mathrm{mp}) \right\rceil,$$

as required. Here, we use the fact that $\lceil \lceil x \rceil /2 \rceil = \lceil x/2 \rceil$ for all $x \in \mathbb{R}$.

Thus, let $H = (G, \mathrm{mp})$ be a multigraph such that $|V(G)| \leq 10$ and $\mathrm{mp}(e)$ is even for all $e \in E(G)$. Suppose for a contradiction that $\chi'(H) \neq \left\lceil \chi'_f(H) \right\rceil$. Thus, by Theorem 1.1 $\chi'(H) \neq \max\{\Delta(H), \lceil t(H) \rceil\}$. Then, it follows from Theorem 2 of [20] that there exists a multigraph $H' = (G', \mathrm{mp}')$ and a vertex $v \in V(G')$ such that (i) G' is isomorphic to the Petersen graph, (ii) H' is regular, (iii) there exists a 5-cycle C in H' that has an odd number of edges, (iv) H is a submultigraph of H', and (v) $H' - v$ is a submultigraph of H. Conditions (iv) and (v) imply that $\mathrm{mp}(e) = \mathrm{mp}'(e)$ for all $e \in E(G) \setminus \delta(v)$. Now let $u \in V(G) \setminus N(v)$ (u exists because G' is isomorphic to the Petersen graph). Since $\mathrm{mp}(e)$ is even for all $e \in \delta(u)$ and H' is regular, it follows that $\deg_{H'}(u)$ is even. Next, consider the 5-cycle C. Clearly, since $\mathrm{mp}(e) = \mathrm{mp}'(e)$ is even for all $e \in E(G) \setminus \delta(v)$, C contains an edge vx such that $\mathrm{mp}'(vx)$ is odd. But because $\mathrm{mp}'(e)$ is even for all $e \in \delta(x) \setminus \{vx\}$, this implies that $\deg_{H'}(x)$ is odd, a contradiction. This proves the theorem. ■

By this theorem all graphs with at most 10 vertices have the lcd-property with constant C either 1 or 2.

5.1 OLoP graphs

In this section, we prove that every OLoP graph has the lcd-property with $C = 1$. Our approach is to prove (ii) in (5.2) with $C = 1$ for any OLoP graph G. That is we need to prove, the following theorem:

Theorem 5.5. *For every* OLoP *multigraph* $H = (G, \mathrm{mp})$, *it holds that* $\chi'(G, \mathrm{mp}) = \left\lceil \chi'_f(G, \mathrm{mp}) \right\rceil$.

Because of (4.5) and (4.9), it suffices to consider blocks of OLoP graphs. Indeed, suppose that Theorem 5.5 holds for blocks and let x be a cut-vertex in an OLoP multigraph H. Let $K^1, \ldots, K^p$ be the connected components of $H - x$. Then,

$$\begin{aligned}\chi'(H) &= \max\left[\deg(x),\ \max_{i=1,\ldots,p} \left\{\chi'(H[V(K^i) \cup \{x\}])\right\}\right] \\ &= \max\left[\deg(x),\ \max_{i=1,\ldots,p} \left\lceil \chi'_f(H[V(K^i) \cup \{x\}]) \right\rceil\right] = \left\lceil \chi'_f(H) \right\rceil.\end{aligned}$$

Thus, we concentrate on the blocks of OLoP graphs. We show that for a multigraph $H = (G, \mathrm{mp})$ such that G is a block of an OLoP graph, it holds that $\chi'(H) = \left\lceil \chi'_f(H) \right\rceil$. The proof relies on the observation that all blocks of an OLoP graph, with the exception of few small blocks with $|V(H)| \leq 5$, are either bipartite or nearly bipartite or can be easily 'reduced' to a nearly bipartite graph. We begin by briefly reviewing the main results for nearly bipartite multigraphs which are of interest to us.

A multigraph $H = (G, \mathrm{mp})$ is called *nearly bipartite* if there exists a vertex $x \in V(G)$ such that $G - x$ is bipartite.

Eggan and Plantholt [6] proved that $\chi'(H) = \max\{\Delta(H), \lceil t(H) \rceil\}$ for every nearly bipartite multigraph H. Thus, by Theorem 1.1 we readily obtain the following.

Theorem 5.6. (Eggan and Plantholt [6]) *If H is a nearly bipartite multigraph, then* $\chi'(H) = \left\lceil \chi'_f(H) \right\rceil$.

Moreover, the following result was shown in [19] (see also [20]).

Theorem 5.7. (Plantholt [19]) *If H is a multigraph with $|V(H)| \leq 8$, then* $\chi'(H) = \left\lceil \chi'_f(H) \right\rceil$.

In the following two results, (5.8) and (5.9), we will use these two theorems to prove Theorem 5.5. We start with the easier case:

(5.8) *Every multigraph H of the $\mathcal{B}_2$ type satisfies* $\chi'(H) = \lceil \chi_f(H) \rceil$.

Proof. Let $H = (G, \mathrm{mp})$ be a multigraph of the $\mathcal{B}_2$ type. First, if G is isomorphic to K_2, K_3 or K_4, then the claim holds by Theorem 5.7. Next, if G is isomorphic to $K_{2,t}$ $(t \geq 2\}$, then H is bipartite and the result follows by Kőnig's theorem [15]. Finally, let G be isomorphic to $K^+_{2,t}$ $(t \geq 2\}$. Let u, v be the two vertices on the side of cardinality 2. Then $H - u$ is isomorphic to $K_{1,t}$. Hence H is nearly bipartite and the result follows from Theorem 5.6. ■

This leaves blocks of the $\mathcal{B}_1$ type:

(5.9) *Every multigraph H of the $\mathcal{B}_1$ type satisfies* $\chi'(H) = \left\lceil \chi'_f(H) \right\rceil$.

Proof. First, consider a multigraph H of the $\mathcal{B}_1$ type with $|V(H)| \geq 6$. Let us start with the case where H is constructed from a 7-cycle.

(i) *If H contains a cycle of length seven, then H is nearly bipartite.*

Consider H and let c_1-c_2-$\cdots$-c_7-c_1 be the vertices of a cycle of length seven in H. From the definition of graphs of the $\mathcal{B}_1$ type, it follows that we may assume that all pairs of vertices c_i, c_j with $|i - j| \geq 2$ are non-adjacent except possibly $\{c_1, c_4\}$, $\{c_1, c_5\}$ and $\{c_4, c_7\}$. If both c_1 and c_4 have clones, then H is a multi-cycle of length seven, and thus the result holds (for instance $H - c_2$ is bipartite). From the symmetry, we may assume now that no vertex is a clone of c_1. We claim that $H - c_1$ is bipartite. Let $C(c_i)$ be the set of clones of vertex c_i, for $i = 2, \ldots, 7$ and let $C[c_i] = C(c_i) \cup \{c_i\}$. Notice that some of the sets $C(c_i)$ are necessarily empty, since only vertices of degree 2 may admit clones. Then the bipartition V_1, V_2 of $H - c_1$ is obtained as follows: $V_1 = \{C[c_3], C[c_5], C[c_7]\}$ and $V_2 = \{C[c_2], C[c_4], C[c_6]\}$. □

By **(i)** and Theorem 5.6, we may assume that H is not nearly bipartite and is constructed from a 5-cycle, say c_1-c_2-$\cdots$-c_5-c_1. If two vertices of the cycle, say c_1 and c_3, admit clones then $H - c_2$

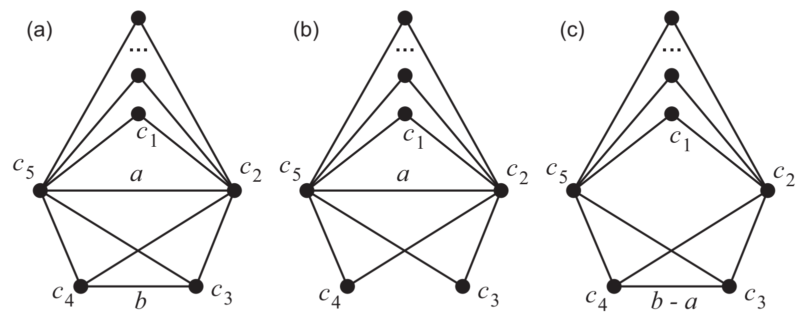

Figure 3: (a) The block H; (b) H' when $a \geq b$; (c) H' when $a < b$

is bipartite and thus H is nearly bipartite, a contradiction. Thus, since $|V(H)| \geq 6$, exactly one vertex of the cycle admits clones. We may assume without loss of generality that c_1 admits clones in H (see Figure 3(a)). Furthermore, the pairs $\{c_2, c_5\}$, $\{c_2, c_4\}$, and $\{c_3, c_5\}$ are adjacent, because otherwise H would be nearly bipartite. Notice that since c_1 admits clones, the pairs $\{c_1, c_3\}$ and $\{c_1, c_4\}$ are non-adjacent. Let $a = \mathrm{mp}(c_2c_5)$ and $b = \mathrm{mp}(c_3c_4)$. We distinguish two cases:

(i) $a \geq b$: Consider the graph H' obtained from H by deleting all edges between c_3 and c_4 (see Figure 3(b)).

(ii) $a < b$: Consider the graph H' obtained from H by deleting all edges between c_2 and c_5 and deleting a edges between c_3 and c_4 (see Figure 3(c)).

We obtain the following result.

(ii) *H' is nearly bipartite and*

$$\chi'(H') = \begin{cases} \chi'(H) & \text{if } a \geq b, \\ \chi'(H) - a & \text{if } a < b, \end{cases} \quad \text{and } \chi'_f(H') = \begin{cases} \chi'_f(H) & \text{if } a \geq b, \\ \chi'_f(H) - a & \text{if } a < b. \end{cases}$$

Suppose that $a \geq b$. Clearly $\chi'(H') \leq \chi'(H)$. Consider now an optimal edge-coloring of H'. Notice that the colors used to color the edges $(c_2c_5)^1, \ldots, (c_2c_5)^a$ are not used to color any other edges in H'. Thus we obtain an edge-coloring of H from the edge-coloring of H' by using the colors of the edges $(c_2c_5)^1, \ldots, (c_2c_5)^a$ to color the edges $(c_3c_4)^1, \ldots, (c_3c_4)^b$. This is possible since $a \geq b$. Thus we obtain a feasible edge-coloring of H using $\chi'(H')$ colors. Hence $\chi'(H') = \chi'(H)$. Clearly H' is nearly bipartite since, for instance, $H' - c_2$ is bipartite.

Now suppose that $a < b$. Consider an optimal edge-coloring of H. By re-coloring some of the edges $(c_3c_4)^1, \ldots, (c_3c_4)^b$, if necessary, we may assume that the edges $(c_2c_5)^1, \ldots, (c_2c_5)^a$ are colored with colors that are also used to color the first a edges of $(c_3c_4)^1, \ldots, (c_3c_4)^b$. These a colors are not used for any other edges in the graph. Now in H', we may assume without loss of generality that exactly those a edges of $(c_3c_4)^1, \ldots, (c_3c_4)^b$ have been deleted. Thus we obtain a feasible edge-coloring of H' with $\chi'(H) - a$ colors. We claim that this coloring is optimal. Indeed, if $\chi'(H') < \chi'(H) - a$, then we would obtain a feasible edge-coloring of H with strictly less than $\chi'(H)$ colors by coloring the edges between vertices c_2 and c_5 as well as the added edges between vertices c_3 and c_4 with a new colors, a contradiction. Clearly H' is nearly bipartite since, for instance, $H' - c_3$ is bipartite.

The proof for the fractional chromatic index of H is similar and thus it is omitted. This proves **(ii)**. □

Since both a and b in the definition of H' are integers, **(ii)** implies that $\chi'(H) = \left\lceil \chi'_f(H) \right\rceil$, proving the lemma for $|V(H)| \geq 6$. Finally, the lemma holds for $|V(H)| = 5$ by Theorem 5.7. This proves the lemma. ■

We just showed that for a multigraph $H = (G, \mathrm{mp})$ such that G is a block of an OLoP graph, it holds that $\lceil \chi'_f(H) \rceil = \chi'(H)$. This allows us to finish up the proof of Theorem 5.5.

Proof of Theorem 5.5. Let $H = (G, \mathrm{mp})$ be an OLoP multigraph. By (4.5), $\chi'(G, \mathrm{mp})$ equals either the maximum cut-vertex degree or the maximum block chromatic index. The same holds for $\chi'_f(G, \mathrm{mp})$. The maximum cut-vertex degree is integral, and by (5.8) and (5.9) we have $\lceil \chi'_f(B) \rceil = \chi'(B)$ for each block of H. This proves that $\lceil \chi'_f(H) \rceil = \chi'(H)$ are required. ■

5.2 GOLoP graphs

In this section, we prove that, for any fixed integer $b \geq 1$, every GOLoP(b) has the lcd-property with constant C, where C only depends on b. The main ingredient for this result is the following lemma.

Lemma 5.10. *Let G be a graph and let (G', W) be the collapsed graph associated with G. Then G has the lcd-property with constant C, where C is a constant that only depends on G' and W.*

Proof. Let G, G', and W be as in the statement of the lemma. Let $p = |W|$. Consider the linear program

$$\min \left\{ \mathbf{e}^T \mathbf{x} \;\middle|\; A\mathbf{x} = \mathbf{b}; \mathbf{x} \geq 0 \right\} \tag{10}$$

where $\mathbf{e}$ is an all-ones vector, A is the constraints matrix of the linear program (9), for G, G', and W, in which the inequality constraints have been replaced by equality constraints with slack variables. We assume that these inequalities constitute the last p rows of A. Finally, the right hand side vector $\mathbf{b}$ is arbitrary. It is well-known that every bounded linear program has an optimal solution that corresponds to a basic feasible solution (see, e.g., [1]). Let

$$C(G', W) = \mathrm{lcm}\left(\det(B) \;\middle|\; B \text{ is a basis matrix for (10)}\right).$$

Notice that $C(G', W)$ exists because there are only finitely many basis matrices for (10). Also observe that $C(G', W)$ only depends on A, and that $C(G', W)$ is independent of the vector $\mathbf{b}$.

Now let r be a given rate function for G such that $\chi'_f(G, r) \leq 1$ and let d be its least common denominator. We will show that $\chi'(G, C(G', W)dr) \leq C(G', W)d$. Following the proof of (4.13), we have that

$$\chi'_f(G, r) = \min \left\{ \mathbf{e}^T \mathbf{x} \;\middle|\; A\mathbf{x} = \begin{pmatrix} \mathbf{r}' \\ \mathbf{t} \end{pmatrix}; \quad \mathbf{x} \geq 0 \right\}, \tag{11}$$

where $\mathbf{r}'$ corresponds to the right-hand-side values in (9) and $\mathbf{t} = \begin{pmatrix} T_1^{**} & \dots & T_p^{**} \end{pmatrix}^T$ where $T_i^{**} = r_i(u_i) + r_i(v_i) - \chi'_f(H_i, r_i)$ and H_i and r_i are as in the proof of (4.13). It follows from the choice of d and the definition of $\mathbf{r}'$ that $d\mathbf{r}'$ is a vector of integers. For $i \in [p]$, it follows from (4.11) that $\chi'_f(H_i, r_i)$, and hence dT_i^{**}, is an integer. Therefore, by scaling the right-hand-side vector of (11), we have

$$d\chi'_f(G,r) = \min\left\{ \mathbf{e}^T\mathbf{x} \;\middle|\; A\mathbf{x} = \begin{pmatrix} d\mathbf{r}' \\ d\mathbf{t} \end{pmatrix}; \quad \mathbf{x} \geq 0 \right\}, \tag{12}$$

which is a linear program in which all entries of both the constraint matrix and the right-hand-side vector are integers. Let B^* be an optimal basis matrix for (12). By Cramer's rule, we have

$$x_i = \frac{\det(B_i^*)}{\det(B^*)},$$

where B_i^* is constructed from B^* by replacing the i'th column by the vector $(d\mathbf{r}', d\mathbf{t})$. Since $(d\mathbf{r}', d\mathbf{t})$ is a vector of integers and the entries of A are all integers, it follows that both the numerator and the denominator in the expression above are integers. Therefore, since $C(G', W)$ is a multiple of $\det(B^*)$, $C(G', W)x_i$ is an integer. Hence, $\mathbf{y} := C(G', W)\mathbf{x}$ is an optimal solution to the integer program

$$C(G',W)d\chi'_f(G,r) = \min\left\{ \mathbf{e}^T\mathbf{y} \;\middle|\; A\mathbf{y} = \begin{pmatrix} C(G',W)d\mathbf{r}' \\ C(G',W)d\mathbf{t} \end{pmatrix}; \quad \mathbf{y} \geq 0; \quad \mathbf{y} \text{ is an integer} \right\}, \tag{13}$$

and the corresponding optimal objective value is $C(G', W)d\chi'_f(G, r)$. Now observe that (13) is exactly the integer program (8) for $\chi'(G, C(G', W)dr)$ in the proof of (4.7). Thus, $\chi'(G, C(G', W)dr) = C(G', W)d\chi'_f(G, r) \leq C(G', W)d$. This proves the lemma. ■

The following theorem uses Lemma 5.10 to establish the lcd-property for all GOLoP(b) graphs with fixed b.

Theorem 5.11. *There exists a function $C(b)$ such that every* GOLoP(b) *graph has the lcd-property with constant at most $C(b)$.*

Proof. Let $C(b) = \mathrm{lcm}(C(G', A))$, where $C(G', A)$ is the constant C from Lemma 5.10 and where the least common denominator is taken over all graphs G' on at most b vertices and A is a stable set of vertices of degree two in G'. Now the result follows from Lemma 5.10, (4.5), the fact that there are only finitely many graphs on b vertices and the fact that for every fixed graph G', there are only finitely many choices of A. ■

5.3 Minimizing the schedule length

After having found a linear (resp. a quadratic) algorithm for the problem K-Match(G, r, k) (resp. Find-Match(G, r, k)), a natural next problem to take on is Min-Match(G, r). That is the problem of finding the smallest k such that the answer to K-Match(G, r, k) is affirmative. While trying to solve this problem, we run into two difficulties. The first difficulty lies in the fact that the smallest value of k might be quite large. The second difficulty follows from the fact that K-MATCH(G, r, k) having an affirmative answer does not necessarily imply that K-MATCH(G, r, k') for $k' > k$ has an affirmative answer. Thus, a straightforward binary search does not work. It seems that the best we can do is a full search of all values of k up to the upper bound that is given by Theorem 5.11. This leads to the following result:

(5.12) *Let $b \geq 1$ be a constant integer. Let G be a* GOLoP(b) *graph and let r be a rate function for G. Then* MIN-MATCH(G, r) *can be solved in $O(|V(G)|d)$ time, where d is the least common denominator of the rates.*

Proof. Let $C = C(b)$ as in Theorem 5.11. Notice that C is a constant. For each block B of G, let $K(B)$ be the set of values $k = 1, \ldots, Cd$ such that there exists a schedule of length k. These sets can be constructed in $O(|V(G)|d)$ total running time. Next, choose the smallest value k such that $k \in K(B)$ for all blocks B, which also takes $O(|V(G)|d)$ time. ∎

Although the algorithm above is efficient in the theoretical sense, we do note that the constant C given by Theorem 5.11 is quite large. However, Conjecture 5.3 suggests that actual constant is as small as 2. Moreover, in the case of OLoP graphs, Theorem 5.5 allows us to use $C = 1$.

6 Conclusions

In this paper, we studied the following problem: given an undirected graph $G = (V, E)$, an integer $k \geq 1$ and an arrival rate $0 < r(e) \leq 1$ for each of its edges $e \in E$, does there exist a sequence of k matchings such that each edge e belongs to at least $kr(e)$ of these matchings? We showed that this problem is NP-complete in the strong sense for general graphs. This result, as well as potential applications in single-hop traffic models of wireless networks with primary interference constraints motivated us to study the problem for a special class of graphs: the GOLoP(b) graphs, a generalization of the so-called OLoP graphs. We presented a linear-time algorithm for deciding whether the abovementioned sequence of matchings exists and a quadratic-time algorithm for finding it for any GOLoP(b) graph with a constant parameter b.

For the problem of finding the smallest value of k such that a sequence of k matchings satisfying the arrival rate constraints exists, we proved that the least common denominator d of the arrival rates is an upper bound for the smallest k for OLoP graphs. At the same time we showed that no schedule shorter than $2d$ may exist for the Petersen graphs. We also conjectured that the upper bound on the length of the shortest schedule is actually $2d$ for general graphs, and we proved this conjecture for general graphs with at most 10 vertices. Finally, we presented a pseudopolynomial time algorithm for the smallest k for GOLoP(b) graphs with constant b. The question of the existence of a polynomial time algorithm for finding the smallest k remains open for OLoP as well as for GOLoP(b) graphs.

While GOLoP(b) graphs are important due to their natural connection to OLoP graphs, it would be interesting to derive the complexity results for more general classes of graphs. This seems to be a promising direction for further research.

Finally, since the potential applications of the results presented in this paper lie in the area of wireless networks where the centralized algorithms are difficult or impossible to implement, an interesting research direction is the investigation of the combinatorial problems considered here in a distributed setting.

Acknowledgments

This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Grant OPG0105675. Moreover, Dariusz Dereniowski has been supported by the Polish Ministry of Science and Higher Education (MNiSW) grant N N516 196437.

References

[1] D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization.* Athena Scientific Belmout, MA, 1997.

[2] B. Birand, M. Chudnovsky, B. Ries, P. Seymour, G. Zussman, and Y. Zwols. Analyzing the performance of greedy maximal scheduling via local pooling and graph theory. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[3] A. Brzezinski, G. Zussman, and E. Modiano. Enabling distributed throughput maximization in wireless mesh networks - a partitioning approach. In *Proc. ACM MOBICOM'06*, Sept. 2006.

[4] A. Dimakis and J. Walrand. Sufficient conditions for stability of longest queue first scheduling: second order properties using fluid limits. *Adv. Appl. Probab.*, 38(2):505–521, June 2006.

[5] J. Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards*, 69(1-2):125–130, 1965.

[6] L. Eggan and M. Plantholt. The chromatic index of nearly bipartite multigraphs. *Journal of Combinatorial Theory, Series B*, 40(1):71–80, 1986.

[7] F. Eisenbrand. Fast integer programming in fixed dimension. In G. Di Battista and U. Zwick, editors, *Proceedings of the 11th Annual European Symposium on Algorithms*, volume 2832 of *LNCS*, pages 196–207. Springer, 2003.

[8] T. Gonzalez and S. Sahni. Open shop scheduling to minimize finish time. *Journal of the ACM (JACM)*, 23(4):665–679, 1976.

[9] J. L. Gross and J. Yellen. *Graph theory and its applications.* CRC press, 2006.

[10] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[11] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34:910–917, 1988.

[12] J.-H. Hoepman. Simple distributed weighted matchings. eprint cs.DC/0410047, Oct. 2004.

[13] I. Holyer. The NP-completeness of edge-colouring. *Siam J. Comput*, 10(4):718–720, 1981.

[14] C. Joo, X. Lin, and N. B. Shroff. Performance limits of greedy maximal matching in multi-hop wireless networks. In *Proc. IEEE CDC'07*, Dec. 2007.

[15] D. Kőnig. Über graphen und ihre anwendung auf determinantentheorie und mengenlehre. *Mathematische Annalen*, 77:453–465, 1916. 10.1007/BF01456961.

[16] A. N. Letchford, G. Reinelt, and D. O. Theis. A faster exact separation algorithm for blossom inequalities. *Integer programming and combinatorial optimization*, pages 19–52, 2004.

[17] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. *IEEE/ACM Trans. Netw.*, 14(2):302–315, Apr. 2006.

[18] M. W. Padberg and M. R. Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7(1):67–80, 1982.

[19] M.J. Plantholt. An order-based upper bound on the chromatic index of a multigraph. *J. Combin. Inform. System Sci.*, 16:271–280, 1991.

[20] M.J Plantholt and S.K. Tipnis. The chromatic index of multigraphs of order at most 10. *Discrete Mathematics*, 177:185–193, 1997.

[21] P. D. Seymour. On multi-colourings of cubic graphs, and conjectures of fulkerson and tutte. In *Proc. London Math. Soc.(3)*, volume 38, pages 423–460. Citeseer, 1979.

[22] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Automat. Contr.*, 37(12):1936–1948, Dec. 1992.

[23] D. B. West et al. *Introduction to graph theory.* Prentice Hall Upper Saddle River, NJ, 2001.