Alcom-FT Technical Report Series
ALCOMFT-TR-01-184

Approximate Constrained Bipartite Edge Coloring*

Ioannis Caragiannis[1], Afonso Ferreira[2], Christos Kaklamanis[1], Stéphane Pérennes[2], Pino Persiano[3], and Hervé Rivano[2]

[1] Computer Technology Institute and
Dept. of Computer Engineering and Informatics,
University of Patras, 26500 Rio, Greece.
E–mail: {caragian,kakl}@cti.gr

[2] MASCOTTE Project, INRIA Sophia Antipolis,
2004 route des Lucioles, B.P. 93,
06902 Sophia Antipolis Cedex, France.
E–mail:{Afonso.Ferreira,Stephane.Perennes,Herve.Rivano}@sophia.inria.fr

[3] Dipartimento di Informatica ed Applicazioni,
Università di Salerno, 84081 Baronissi, Italy.
E–mail: giuper@dia.unisa.it

Abstract. We study the following *Constrained Bipartite Edge Coloring (CBEC)* problem: We are given a bipartite graph $G(U, V, E)$ of maximum degree l with n vertices, in which some of the edges have been legally colored with c colors. We wish to complete the coloring of the edges of G minimizing the total number of colors used. The problem has been proved to be NP–hard even for bipartite graphs of maximum degree three [5]. In previous work Caragiannis et al. [2] consider two special cases of the problem and proved tight bounds on the optimal number of colors by decomposing the bipartite graph into matchings which are colored into pairs using detailed potential and averaging arguments. Their techniques lead to 3/2–aproximation algorithms for both problems. In this paper we present a randomized $(1.37 + o(1))$–approximation algorithm for the general problem in the case where $\max\{l, c\} = \omega(\ln n)$. Our techniques are motivated by recent work of Kumar [11] on the *Circular Arc Coloring* problem and are essentially different and simpler than those presented in [2].

1 Introduction

König's classical result from graph theory [10], states that the edges of a bipartite graph with maximum degree l can be colored using exactly l colors so that edges that share an endpoint are assigned different colors (see also [1]). We call

* This work was partially funded by the European Communities under IST FET Project ALCOM–FT and RTN Project ARACNE.

such edge colorings *legal* colorings. König's proof [10] is constructive, yielding a polynomial–time algorithm for finding optimal bipartite edge colorings. Faster algorithms have been presented in [3, 6, 7, 15]. These algorithms usually use as a subroutine an algorithm that finds perfect matchings in bipartite graphs [8, 15].

Bipartite edge coloring can be used to model scheduling problems such as timetabling. An instance of timetabling consists of a set of teachers, a set of classes, and a list of pairs (t, c) indicating that the teacher t has to teach class c during a time slot within the time span of the schedule [15]. A timetable is an assignment of the pairs to time slots in such a way that no teacher t and no class c occurs in two pairs that are assigned to the same time slot. This problem can be modelled as an edge coloring problem on a bipartite graph.

In real–life situations, the problem is made somehow harder due to additional constraints that are imposed on the solutions. This is a general feature of practical optimization problems and it is due to the fact that an optimization problem at hand is most of the time just a subproblem of a larger–scale optimazition that one seeks to obtain. In the example of scheduling classes and teachers, it is sometimes the case that some teachers have been assigned to a timeslot because of some other duties that they have to attend during other time slots; thus, assignments will have to take into account this extra restriction. So, usually, additional constraints that are put on a timetable make the problem NP–complete [4].

Caragiannis et al. in [2] study two special cases of problem CBEC that arise from algorithmic problems in optical networks (see [12, 9]). Their results can be summarized as follows:

- **Problem A:** Some of the edges adjacent to a specific pair of opposite vertices of an l–regular bipartite graph are already colored with S colors that appear only on one edge (*single* colors) and D colors that appear on two edges (*double* colors). They show that the rest of the edges can be colored using at most $\max\{\min\{l + D, \frac{3l}{2}\}, l + \frac{S+D}{2}\}$ total colors. They also show that this bound is tight by constructing instances in which $\max\{\min\{l + D, \frac{3l}{2}\}, l + \frac{S+D}{2}\}$ colors are indeed necessary.
- **Problem B:** Some of the edges of an l–regular bipartite graph are already colored with S colors that appear only on one edge. They show that the rest of the edges can be colored using at most $\max\{l + S/2, S\}$ total colors. They also show that this bound is tight by constructing instances in which $\max\{l + S/2, S\}$ total colors are necessary.

Their techniques are based on the decomposition of the bipartite graph into matchings which are colored into pairs using detailed potential and averaging arguments. Their results imply 3/2–aproximation algorithms for both problems.

The original proofs in [2] consider l–regular bipartite graphs $G(U, V, E)$ with $|U| = |V| = n/2$. However, these results extend to bipartite graphs of maximum degree l with n vertices using a simple observation presented in Section 2. Note that CBEC has been proved to be NP–hard even for bipartite graphs of maximum degree three [5].

Our approach. In this paper, motivated by recent work of Kumar [11] on the circular arc coloring problem, the steps we follow to obtain a provably good approximation to problem CBEC are summarized below:

- Given a bipartite graph of maximum degree l in which some of the edges are legally colored with c colors, we reduce the problem to an integral multicommodity flow problem with constraints.
- We formulate the multicommodity flow problem as a 0–1 integer linear program.
- We relax the integrality constraint, and solve the linear programming relaxation obtaining an optimal fractional solution.
- We use randomized rounding to obtain a provably good integer solution of the integral multicommodity flow problem which corresponds to a partial edge coloring.
- We extend the edge coloring by assigning extra colors to uncolored edges.

In this way we extend the coloring of the edges of G using a total number of colors which is provably close to the optimal one. Our algorithm is randomized and works with high probability provided that the optimal number of colors is large (i.e., $\omega(\log n)$).

Roadmap. The rest of the paper is structured as follows. We present the reduction from the costrained bipartite edge coloring problem to an integral multicommodity flow problem in Section 2. In Section 3 we demostrate how to approximate the solution of the integral multicommodity flow problem and prove that this solution corresponds to an approximate edge coloring. An improvement to our approach is presented in Section 4.

2 Bipartite edge coloring and multicommodity flows

In this section we describe the reduction of an instance of problem CBEC to an instance of an integral multicommodity flow problem with constraints. We first present a reduction of the initial instance of the CBEC problem to the following one.

Let $G = (U, V, E)$ be a bipartite graph with $n = n_1 + n_2$ vertices, with $U = \{u_1, ..., u_{n_1}\}$, $V = \{v_1, ..., v_{n_2}\}$, and with maximum degree l, in which some of the edges in E are already legally colored. For any integer $k \geq 0$, we construct the bipartite graph $G_k = (A, B, E(G_k))$ where the sets of vertices A and B are defined as

$$A = \{x_i | u_i \in U\} \cup \{y'_i | v_i \in V\},$$

and

$$B = \{y_i | v_i \in V\} \cup \{x'_i | u_i \in U\}.$$

For graph G_0, the set of edges $E(G_0)$ is defined as follows. For any edge $(u_i, v_j) \in E(G)$ with $u_i \in U$ and $v_j \in V$, $E(G_0)$ contains two edges: (x_i, y_j) and (x'_i, y'_j). We call these edges *regular* edges. Also, let l be the maximum degree of G and let $d(u_i)$ (resp. $d(v_i)$) be the degree of a vertex $u_i \in U$ (resp. $v_i \in V$) in G. The edge set $E(G_0)$ also contains $l - d(u_i)$ copies of (x_i, x'_i) for $i = 1, ..., n_1$, and $l - d(v_i)$ copies of (y_i, y'_i) for $i = 1, ..., n_2$. These edges are called *cross* edges. Graph G_k for $k \geq 0$ is obtained from G_0, by adding k copies of the edges (x_i, x'_i) for $i = 1, ..., n_1$, and k copies of the edges (y_i, y'_i), for $i = 1, ..., n_2$. An example for the construction of graph G_k from G is depicted in Figure 1.

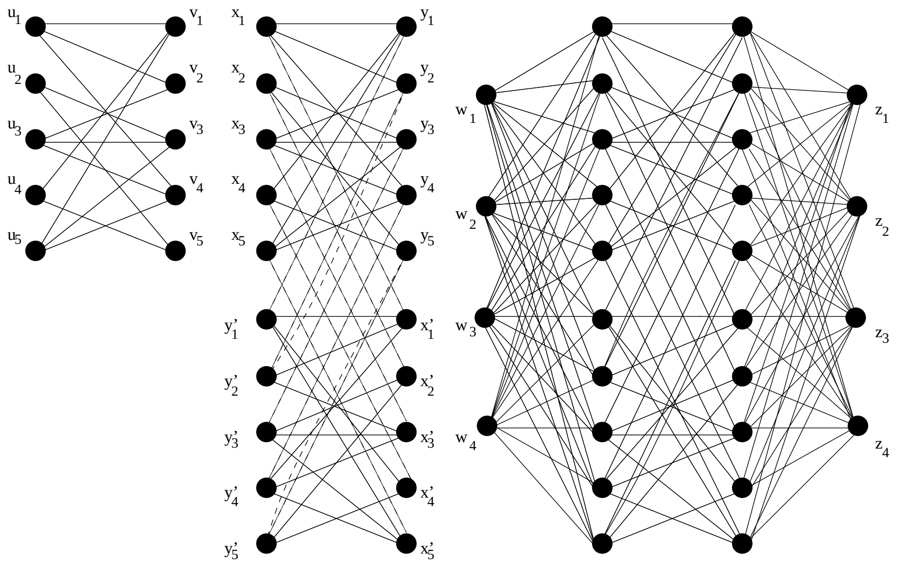

Fig. 1. The graph G, the graph G_1, and the corresponding multicommodity network H_1.

Lemma 1. *G can be edge colored with $l + k$ colors iff G_k can be edge colored with $l + k$ colors.*

Proof. Since G is a subgraph of G_k, any legal edge coloring of G_k trivially yields a legal edge coloring of G.

Assume that we have a legal edge coloring of G with $l + k$ colors. Then, the edges of G_k can be colored with $l + k$ colors as follows. For any edge $(u_i, v_j) \in E(G)$ colored with a color χ, we color the edges (x_i, y_j) and (x'_i, y'_j) of $E(G_k)$ with χ. This gives a partial edge coloring of G_k in which the cross edges are uncolored. Let $u_i \in U$ (resp. $v_i \in V$) and let $d(u_i)$ (resp. $d(v_i)$) be the degree of u_i (resp. v_i) in G. The cross edges between x_i and x'_i (resp. between y_i and y'_i) are now constrained by $d(u_i)$ (resp. $d(v_i)$) colors. Thus, we can use the $l + k - d(u_i)$ (resp. $l + k - d(v_i)$) colors not used by edges adjacent to u_i (resp. v_i) to color the cross edges between x_i and x'_i (resp. y_i and y'_i). This completes the coloring of G_k with $l + k$ colors. $\square$

Now, for any integer $k \geq 0$, consider the multicommodity network $H_k = (W, A, B, Z, E(H_k))$ constructed as follows. Sets of vertices A and B are the same with those of graph G_k. Also,

$$W = \{w_1, ..., w_{l+k}\}$$

and

$$Z = \{z_1, ..., z_{l+k}\}.$$

The set $E(H_k)$ is defined as

$$\begin{aligned} E(H_k) = E(G_k) & \\ & \cup\{(w_i, x_j) | 1 \leq i \leq l+k, 1 \leq j \leq n_1\} \\ & \cup\{(w_i, y'_j) | 1 \leq i \leq l+k, 1 \leq j \leq n_2\} \\ & \cup\{(y_j, z_i) | 1 \leq i \leq l+k, 1 \leq j \leq n_2\} \\ & \cup\{(x'_j, z_i) | 1 \leq i \leq l+k, 1 \leq j \leq n_1\} \end{aligned}$$

All the edges in $E(H_k)$ have unit capacity, and an edge can carry only an integral amount of flow for each commodity. The source for the i–th commodity is located at w_i, while the corresponding sink is located at z_i. An example for the construction of network H_k from graph G_k is depicted in Figure 1.

Intuitively, an integral flow of the $l+k$ commodities corresponds to a (partial) legal coloring of the edges of G_k: an edge between A and B carrying one unit of flow for commodity i in H_k corresponds to an edge colored with color i in G_k.

Since some of the the edges of the graph G_k are precolored, our multicommodity flow problem has some additional constraints. If an edge is precolored with color i in G_k, it is constrained to carry a unit amount of flow for commodity i in H_k. So, we can reduce an instance of CBEC to multicommodity flow with constraints using the following observation.

Lemma 2. *G_k can be edge colored with $l + k$ colors iff there is an integral (constrained) flow of value $n(l + k)$ for commodities $1, ..., l + k$ in network H_k.*

In the next section we show how to approximate the corresponding integral constrained multicommodity flow problem, and, using the reduction above, we obtain a provably good solution for the initial instance of CBEC.

3 Approximating the multicommodity flow problem

In general, integral multicommodity flow (without constraints) is NP–complete [4]. However, it is straightforward to formulate the constrained multicommodity flow problem as a 0–1 integer linear program and solve its linear programming relaxation by setting aside the integrality constraint. In this way, we obtain an optimal fractional solution.

Clearly, $\max\{l, c\}$ is a lower bound on the minimum number of colors sufficient for extending the partial edge–coloring of G. We begin with network

$H_{\max\{l,c\}-l}$, solving the corresponding linear program $\mathrm{LP}_{\max\{l,c\}-l}$. If the maximum flow is smaller than $n\max\{l,c\}$, this means that the integer linear program has no flow with value $n\max\{l,c\}$, meaning (by Lemma 2) that there exist no coloring of $G_{\max\{l,c\}-l}$ with $\max\{l,c\}$ colors. We continue with networks $H_{\max\{l,c\}-l+1}, H_{\max\{l,c\}-l+2}, \ldots$, until we find some L such that the solution of LP_{L-l} gives a fractional (constrained) multicommodity flow of value nL. Clearly, L is a lower bound for the minimum number of colors sufficient for coloring the edges of G_{L-l}.

Now, we will use the fractional solution of the linear program LP_{L-l} in order to obtain a solution for the corresponding integer linear program ILP_{L-l} which is provably close to the optimal one. We will use the randomized rounding technique proposed by Raghavan [14].

Let f be the flow obtained by solving LP_{L-l}. Flow f can be decomposed into L flows $f_1, f_2, \ldots, f_L$; one for each commodity. Each f_i can be further broken up into sets $P_{i,1}$, $P_{i,2}$, ..., of n vertex–disjoint paths from w_i to z_i (i.e., the edges between A and B in each set of vertex–disjoint paths forms a perfect matching) each carrying an amount $m_{i,j}$ of flow for commodity i, such that $\sum_{j=1}^{t_i} m_{i,j} = f_i = 1$. We call the procedure of decomposing flow *matching stripping* (since it is similar in spirit to the path stripping technique proposed in [14]).

Lemma 3. *Matching stripping can be done in polynomial time.*

Proof. Matching stripping can be performed as follows. Consider a solution to LP_k and the associated flows for commodity i in network H_k. Set $j=1$. Let e_j be the edge carrying the smallest non–zero amount $m_{i,j}$ of flow for commodity i. Find a set $P_{i,j}$ of n vertex–disjoint paths from w_i to z_i containing edges that carry non–zero amount of flow for commodity i including e_j. Associate amount $m_{i,j}$ with $P_{i,j}$ and subtract amount $m_{i,j}$ from the flow for commodity i carried by each edge in $P_{i,j}$. Repeat this process for $j = 2, 3, \ldots$, until no flow remains. This will decompose the flow f_i into sets of n vertex–disjoint paths $P_{i,j}$ between w_i and z_i each carrying amount $m_{i,j}$ of flow for commodity i.

We first inductively prove that a set of n vertex–disjoint paths $P_{i,j}$ from w_i to z_i can be found at any execution of the above process. Let e_1 be the edge carrying the smallest non–zero amount $m_{i,1}$ of flow for commodity i in the beginning of the first execution. Assume that any set of vertex–disjoint paths from w_i to z_i containing edges that carry non–zero amount of flow for commodity i including e_j has size at most $n-1$. This means that there is no perfect matching containing e_1 in the subgraph of H_k containing the vertex sets A and B and the edges between them that carry non–zero amount of flow for commodity i. By Hall's Matching Theorem (see [1]), we obtain that there exists a set $S \subseteq A$ (such that e_1 is incident to one of its vertices) with neighborhood $N(S) \subseteq B$ of size $|N(S)| \leq |S| - 1$. Observe that since the solution of LP_k is optimal the edges incident to a vertex of A carry unit total amount of flow for commodity i. Thus, the edges incident to S carry a total amount $|S|$ of flow for commodity i and, since $|N(S)| \leq |S| - 1$, the capacity constraints for some of the edges incident to $N(S)$ are violated. Thus, a perfect matching $M_{i,1}$ containing edges between

A and B including e_1 exists. The set $P_{i,1}$ of vertex–disjoint paths is constructed by adding all edges between w_i and A and all edges between B and z_i to $M_{i,1}$.

Assume now that $j-1$ sets of n vertex disjoint paths $P_{i,1}$, $P_{i,2}$, ..., $P_{i,j-1}$ between w_i and z_i have been constructed in the beginning of the j–th execution of the above process and let $m_{i,1}$, $m_{i,2}$, ..., $m_{i,j-1}$ be the associated flows for commodity i. Furthermore, assume that there still exists an edge which carries non–zero amount of flow for commodity i. Note that an amount of $\sum_{t=1}^{j-1} m_{i,t}$ of flow for commodity i has been subtracted from each edge between w_i and A, from each edge between B and z_i, from the edges between A and B incident to each vertex of A, and, similarly, from the edges between A and B incident to each vertex of B. Following the same reasoning as above, we consider the edge e_j carrying the smallest non–zero amount of flow and we obtain that there exists a perfect matching $M_{i,j}$ between A and B containing edges that carry non–zero amount of flow for commodity i including e_j (otherwise, some the edge capacity contraints in the original solution of LP_k would have been violated). Again, the set $P_{i,j}$ of vertex–disjoint paths is constructed by adding all edges between w_i and A and all edges between B and z_i to $M_{i,j}$.

We now easily prove that the number t_i of executions of the above process is polynomial. Observe that after the j–th execution, there exists at least one edge (e_j) which carry zero amount of flow, and, thus, it will not be considered in the construction of paths $P_{i,t}$ for $t > j$. Thus, the number of executions of the process is at most the number of edges between A and B, i.e., $t_i \leq n(l+k)$.

The lemma follows since maximum bipartite matching can be solved in polynomial time. □

In order to obtain an integer solution for ILP_{L-l}, for each commodity i, we will select one out of the t_i sets of vertex–disjoint paths, and use its edges to route commodity i. To select a set of vertex–disjoint paths for commodity i, we cast a t_i–faced die (one face per each of the t_i sets of vertex–disjoint paths) where $m_{i,j}$ are the probabilities associated with the faces. The selection is performed independently for each commodity. Performing this procedure for each commodity, we obtain L sets of n vertex–disjoint paths to route the L commodities.

However, these sets of n vertex–disjoint paths may not constitute a feasible integer solution to ILP_{L-l} since some edge capacities may be violated. Since in the fractional solution an edge between A and B may carry more than one commodity, it is possible that, during the rounding procedure, more than one commodities may select sets of vertex–disjoint paths that contain that edge.

Next, in each edge between A and B that was selected by more than one commodities we arbitrarily select one commodity that will use this edge. In this way, we obtain a feasible integer solution for ILP_{L-l}.

Note that the feasible solution of the integral multicommodity flow problem in H_{L-l} corresponds to a partial edge coloring of G_{L-l} with L colors. We also have to assign extra colors to the edges that do not belong to the sets of vertex–disjoint paths that were selected by the rounding procedure. Let G'_{L-l} be the (random) subgraph of G_{L-l} that contains all vertices of G_{L-l} and the edges that do not correspond to edges of H_{L-l} that were selected by the rounding procedure.

Next, in Lemma 5, we will provide an upper bound on the maximum degree of graph G'_{L-l}. Our proof is based on the following technical lemma on a well–known occupancy problem. A proof can be found in Kumar [11] (see also [13]).

Lemma 4. *Consider the process of randomly throwing m_1 balls into m_2 bins such that the expectation of the number of balls thrown into any bin is at most one. For the random variable Z denoting the number of empty bins, it holds that*

$$\Pr[Z \geq m_2 - m_1 + m_1/e + \lambda\sqrt{m_1}] \leq 2\exp(-\lambda^2/2).$$

Lemma 5. *The maximum degree of G'_{L-l} is at most $L/e + 2\sqrt{L \ln n}$, with probability at least $1 - 4/n$.*

Proof. During randomized rounding, each commodity i randomly selects a set of n vertex–disjoint paths between the source w_i and destination z_i in H_{L-l}. Thus, for each node u of G_{L-l}, one of the L edges incident to u is selected to carry unit flow for a specific commodity. Intuitively, we can think of the integral flow for each commodity as a ball and the edges incident to a vertex u as bins. The randomized rounding procedure can be modelled by the classical occupancy problem where L balls are to be randomly and independently thrown into L bins with the restriction that the expectation of the number of balls thrown into any bin is at most one (this is due to the edge capacity constraints of the flow problem). Using Lemma 4 with $m_1 = m_2 = L$ and $\lambda = 2\sqrt{\ln n}$, we obtain that the random variable denoting the number of empty bins, i.e., the number of edges incident to u which are not selected for carrying flow for any commodity, is at most $L/e + 2\sqrt{L \ln n}$ with probability at least $1 - 2/n^2$.

Thus, the probability that more than $L/e+2\sqrt{L \ln n}$ edges incident to some of the $2n$ vertices of G_k have not been selected after the execution of the randomized rounding procedure is at most $2n \cdot 2/n^2 = 4/n$. The lemma follows. □

By Lemma 5, the edges of G'_{L-l} can be colored with at most $L/e + 2\sqrt{L \ln n}$ extra colors, with high probability. Thus, in the case where L is large (i.e., $L = \omega(\ln n)$), we have proved the following theorem.

Theorem 1. *With very high probability, the algorithm uses at most $(1+1/e)L + o(L)$ total colors.*

Since L is a lower bound to the optimal number of colors sufficient for coloring the edges of the bipartite graph, we obtain that our algorithm has approximation ratio $1 + 1/e + o(1) = 1.37 + o(1)$.

4 Decreasing the number of colors

In this section we discuss some modifications of our algorithm which lead to a better upper bound on the total number of colors sufficient for solving instances of CBEC. Note that this improved result does not imply an approximation ratio better than the one obtained in Section 3.

We slightly modify the reduction described in Section 2. Consider again the bipartite graphs $G_k = (A, B, E(G_k))$ (for integer $k \geq 0$) defined in Section 2. For any integer $k \geq \max\{l, c\} - l$, we construct the multicommodity flow network $H'_k = (W', A, B, Z', E(H'_k))$ where now

$$W' = \{w_1, ..., w_{\max\{l,c\}}\}$$

and

$$Z' = \{z_1, ..., z_{\max\{l,c\}}\}.$$

The set $E(H'_k)$ is defined as

$$\begin{aligned} E(H'_k) = E(G_k) & \\ & \cup\{(w_i, x_j) | 1 \leq i \leq \max\{l, c\}, 1 \leq j \leq n_1\} \\ & \cup\{(w_i, y'_j) | 1 \leq i \leq \max\{l, c\}, 1 \leq j \leq n_2\} \\ & \cup\{(y_j, z_i) | 1 \leq i \leq \max\{l, c\}, 1 \leq j \leq n_2\} \\ & \cup\{(x'_j, z_i) | 1 \leq i \leq \max\{l, c\}, 1 \leq j \leq n_1\} \end{aligned}$$

Our reduction is now based on the following lemma.

Lemma 6. *G_k can be edge colored with $l + k$ colors iff there is an integral (constrained) flow of value $n \max\{l, c\}$ for commodities in network H'_k.*

Proof. A coloring of G_k with $l + k$ colors can be reduced to an integral (constrained) flow of value $n \max\{l, c\}$ for commodities in network H'_k by making each edge between A and B colored with some color i in G_k (for $1 \leq i \leq \max\{l, c\}$) carry a unit amount of flow for commodity i.

Given an integral (constrained) flow of value $n \max\{l, c\}$ for commodities in network H'_k, we can achieve a partial coloring of G_k with $\max\{l, c\}$ colors by using color i (for $1 \leq i \leq \max\{l, c\}$) to color an edge which carries a unit amount of flow for commodity i. We observe that the vertex–induced subgraph of G_k which contains the edges of G_k left uncolored is $(l+k-\max\{l, c\})$–regular. Thus, $l + k - \max\{l, c\}$ colors can be used to complete the coloring of the edges of G_k with $l + k$ colors in total. □

The general structure of our approach is the same with the one described in Section 3. We begin with network $H'_{\max\{l,c\}-l}$, solving the corresponding linear program $\mathrm{LP}_{\max\{l,c\}-l}$. If the maximum flow is smaller than $n\max\{l, c\}$, this means that the integer linear program has no flow with value $n\max\{l, c\}$, meaning (by Lemma 6) that there exist no coloring of $G_{\max\{l,c\}-l}$ with $\max\{l, c\}$ colors. We continue with networks $H'_{\max\{l,c\}-l+1}, H'_{\max\{l,c\}-l+2}, ...$, until we find some L such that the solution of LP_{L-l} gives a fractional (constrained) multicommodity flow of value $n \max\{l, c\}$. By Lemma 6, L is a lower bound for the minimum number of colors sufficient for coloring the edges of G_{L-l}.

Then, we use the fractional solution of LP_{L-l} to obtain a feasible solution of ILP_{L-l} using randomized rounding. Again, we can prove that matching stripping can be correctly performed in polynomial time; however, some minor modifications are needed in the proof of Lemma 3.

In order to obtain an upper bound on the degree of the graph G'_{L-l} (the subgraph of G_{L-l} containing edges of G_{L-l} left uncolored after the application of the rounding procedure), we again use Lemma 4 (with $m_1 = \max\{l, c\}$ and $m_2 = L$) to show that G'_{L-l} can be edge colored with $L - \max\{l, c\} + \max\{l, c\}/e + o(\max\{l, c\})$ additional colors. In this way, when $\max\{l, c\} = \omega(\ln n)$, we obtain the following.

Theorem 2. *With very high probability, the algorithm uses at most* $L + \frac{\max\{l,c\}}{e} + o(\max\{l, c\})$ *total colors.*

References

1. C. Berge. Graphs and Hypergraphs. *North Holland*, 1973.
2. I. Caragiannis, C. Kaklamanis, and P. Persiano. Edge Coloring of Bipartite Graphs with Constraints. In *Proc. of the 24th International Symposium on Mathematical Foundations of Computer Science (MFCS 99)*, LNCS 1672, Springer, pp. 771–779, 1999.
3. R. Cole and J. Hopcroft. On Edge Coloring Bipartite Graphs. *SIAM Journal on Computing*, 11 (1982), pp. 540–546.
4. S. Even, A. Itai, and A. Shamir. On the Complexity of Timetable and Multicommodity Flow Problems. *SIAM Journal on Computing*, 5 (1976), pp. 691–703.
5. J. Fiala. NP–Completeness of the Edge Precoloring Extension Problem on Bipartite Graphs. Manuscript, 2000.
6. H.N. Gabow. Using Eulerian Partitions to Edge Color Bipartite Graphs. *Internat. J. Comput. Inform. Sci.*, 5 (1976), pp. 345–355.
7. H.N. Gabow and O. Kariv. Algorithms for Edge Coloring Bipartite Graphs and Multigraphs. *SIAM Journal on Computing*, 11 (1982), pp. 117–129.
8. J. Hopcroft and R. Karp. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, 2 (1973), pp. 225–231.
9. C. Kaklamanis, P. Persiano, T. Erlebach, K. Jansen. Constrained Bipartite Edge Coloring with Applications to Wavelength Routing. In *Proc. of the 24th International Colloquium on Automata, Languages, and Programming (ICALP '97)*, LNCS 1256, Springer Verlag, 1997, pp. 493–504.
10. D. König. Graphok és alkalmazásuk a determinások és a halmazok elméletére. *Mathematikai és Természettudományi Értesitö*, 34 (1916), pp. 104–119 (in Hungarian).
11. V. Kumar. Approximating Circular Arc Colouring and Bandwidth Allocation in All–Optical Ring Networks. In *Proc. of the 1st International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX '98)*, LNCS, Springer, 1998. Journal version: An Approximation Algorithm for Circular Arc Coloring. Algorithmica, to appear.
12. M. Mihail, C. Kaklamanis, S. Rao. Efficient Access to Optical Bandwidth. In *Proc. of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95)*, 1995, pp. 548–557.

13. R. Motwani and P. Raghavan. Randomized Algorithms. Cambridge University Press, 1995.
14. P. Raghavan. Randomized Rounding and Discrete Ham–Sandwiches: Provably Good Algorithms for Routing and Packing Problems. PhD Thesis, UC Berkeley, 1986.
15. A. Schrijver. Bipartite Edge Coloring in $O(\Delta m)$ Time. *SIAM Journal on Computing*, 28(3) (1998), pp. 841–846.