# Sum Edge Coloring of Multigraphs via Configuration LP

MAGNÚS M. HALLDÓRSSON
Reykjavik University
GUY KORTSARZ
Rutgers University
and
MAXIM SVIRIDENKO
IBM T. J. Watson Research Center

We consider the scheduling of biprocessor jobs under sum objective (BPSMS). Given a collection of unit-length jobs where each job requires the use of two processors, find a schedule such that no two jobs involving the same processor run concurrently. The objective is to minimize the sum of the completion times of the jobs. Equivalently, we would like to find a sum edge coloring of a given multigraph, i.e. a partition of its edge set into matchings $M_1, \ldots, M_t$ minimizing $\sum_{i=1}^{t} i|M_i|$.

This problem is APX-hard, even in the case of bipartite graphs [Marx 2009]. This special case is closely related to the classic open shop scheduling problem. We give a 1.8298-approximation algorithm for BPSMS improving the previously best ratio known of 2 [Bar-Noy et al. 1998]. The algorithm combines a configuration LP with greedy methods, using non-standard randomized rounding on the LP fractions. We also give an efficient combinatorial 1.8886-approximation algorithm for the case of simple graphs, which gives an improved $1.79568 + O(\log \overline{d}/\overline{d})$-approximation in graphs of average degree $\overline{d}$.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Sequencing and scheduling*; G.1.6 [**Numerical Analyais**]: Optimization—*Linear programming*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Edge Scheduling, Configuration LP, Approximation Algorithms

---

A preliminary version of the article was presented at the 13th Conference on Integer Programming and Combinatorial Optimization (IPCO '08).
M. M. Halldórsson's research was partially supported by by Icelandic Research Fund grant no. 60034022. G. Kortsarz's research was partially supported by NSF Award 072887.
Authors' addresses: M. M. Halldórsson, School of Computer Science, Reykjavik University, 103 Reykjavik, Iceland. e-mail: `mmh@ru.is`; G. Kortsarz, Department of Computer Science, Rutgers University, Camden, NJ 08102. Work done partly while visiting IBM T. J. Watson Research Center. Supported in part by NSF Award 072887. e-mail: `guyk@camden.rutgers.edu`; M. Sviridenko, IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598. e-mail: `sviri@us.ibm.com`

## 1. INTRODUCTION

We consider the problem of *biprocessor scheduling* unit jobs under sum of completion times measure (BPSMS). Given a collection of unit-length jobs where each job requires the use of two processors, find a schedule such that jobs using the same processor never run concurrently. The objective is to minimize the sum of the completion times of the jobs.

The problem can be formalized as an edge coloring problem on multigraphs, where the nodes correspond to processors and edges correspond to jobs. In each round, we schedule a matching from the graph with the aim of scheduling the edges early on average. This is also known as *sum edge coloring*. Formally, the BPSMS problem is: given a multigraphs $G$, find an edge coloring, or a partition into a matchings $M_1, M_2, \ldots$, minimizing $\sum_i i \cdot |M_i|$.

Biprocessor scheduling problems have been studied extensively. Natural applications of biprocessor scheduling include, e.g., data migration problems, [Giaro et al. 2002; Coffman, Jr et al. 1985; Kim 2005], running two (or more) programs on the same job in order to assure that the result is reliable [Gehringer et al. 1986], mutual testing of processors in biprocessor diagnostic links [Krawczyk and Kubale 1985], and batch manufacturing where jobs simultaneously require two resources for processing [Dobson and Karmarkar 1989].

We survey some additional previous work on biprocessor scheduling. If jobs have varying length then biprocessor scheduling of trees is NP-hard both in preemptive [Marx 2005] and non-preemptive [Kubale 1996] settings. Marx [2006] designed a polynomial-time approximation scheme (PTAS) for the case of preemptive biprocessor scheduling of trees. This PTAS was generalized in [Marx 2004] to partial $k$-trees and planar graphs. Marx [2009] showed that BPSMS (unit jobs) is NP-hard even in the case of planar bipartite graphs of maximum degree 3 and APX-hard in general bipartite graphs. When the number of processors is constant, Afrati *et al.* [2000] gave a polynomial time approximation scheme for minimizing the sum of completion times of biprocessor scheduling. Later, their results were generalized in [Fishkin et al. 2001] to handle weighted completion times and release times.

1.0.0.1 *Related problems:.* A problem related to ours is minsum edge coloring with total vertex completion time objective. This problem has a setting similar to ours but with different objective function: for each vertex compute the maximal color assigned to it and minimize the total sum of such colors over all vertices. This problem is NP-hard but admits an approximation ratio of $1+\phi \approx 2.61$ [Mestre 2008] on general graphs.

A problem more general than ours is the (vertex) *sum coloring* problem. Here, a feasible solution assigns a color (a positive integer) to each vertex of the input graph so that adjacent vertices get different colors. The goal is to minimize the sum of colors over all vertices. Thus, the BPSMS problem is the special case of sum coloring *line graphs* of general graphs.

We survey some of the work done on the sum coloring problem. This problem is NP-hard on interval graphs [Marx 2005], planar graphs [Halldórsson and Kortsarz 2002], and is APX-hard on bipartite graphs [Bar-Noy and Kortsarz 1998]. It admits a 1.796-approximation on interval graphs [Halldórsson et al. 2003], a PTAS on planar graphs [Halldórsson and Kortsarz 2002] and a 27/26-approximation on bipartite graphs [Malafiejski et al. 2004].

The literature on scheduling with the total completion time objective is quite extensive. The approximation algorithms for such problems are usually based on completion time formulation [Hall et al. 1997] or time-indexed formulation [Schulz and Skutella 2002]. We refer the reader to [Chekuri and Khanna 2004] for a thorough survey of this subject.

The only general approximation result for BPSMS is a 2-approximation algorithm in [Bar-Noy et al. 1998]. Improved bounds of 1.796 [Gandhi et al. 2008] and $\sqrt{2}$ [Gandhi and Mestre 2009] have been given for the special case of bipartite graphs. The bipartite case lends additional importance to BPSMS as it captures a version of the classic open shop scheduling problem and can be denoted as $O|p_{ij} = 1|\sum C_{ij}$ using the standard scheduling notation [Lawler et al. 1993]. The minsum open shop scheduling with general processing times was studied in the series of papers [Queyranne and Sviridenko 2002a; 2002b; Gandhi et al. 2008].

Our main tool for the main result is the solution of a *configuration linear program (LP)*. A configuration LP is a linear program with a large (usually exponential) number of variables. Usually, it contains a variable corresponding to each "restricted" feasible solution, which in our case corresponds to a feasible assignment of edges to a particular color. See also configuration LP's for assignment problems [Nutov et al. 2006; Bansal and Sviridenko 2006; Chuzhoy et al. 2006] and an interesting application of the technique giving a variable for any tree spanning a certain collection of terminals [Charikar et al. 1998]. Although we are not aware of the instances with the integrality gap close to our performance guarantee even for simpler linear program with assignment variables $x_{et}$ for each edge $e$ and color $t$ it easy to see that the configuration LP is strictly stronger than the simple LP. For a multigraph consisting of three vertices and $k$ parallel edges between each pair of vertices the simple LP has optimal value equal to $3k(k + 1/2)$ (just spread each edge equally between $2k$ colors) while the value of an optimal integral solution and the optimal value of the configuration LP is equal to $3k(3k + 1)/2$.

## 1.1 Our results

THEOREM 1.1. *The BPSMS problem admits an LP-based approximation algorithm with expected approximation ratio at most* 1.8298.

This results holds even if the graph has parallel edges.

This LP-based approach has high polynomial time complexity. Hence, it is of practical interest to study purely combinatorial algorithms.

THEOREM 1.2. *The BPSMS problem on graphs with no parallel edges admits a combinatorial algorithm whose approximation ratio is at most* 1.8886.

This algorithm combines ideas from [Halldórsson et al. 2003] and [Vizing 1964]. In the case of non-sparse graphs, it also leads to a ratio that improves on the LP-based approach, or $1.79568 + O(\log \overline{d}/\overline{d})$, where $\overline{d}$ is the average degree. Note that this result gives better ratio than the LP if $\bar{d}$ is large enough. But it works only for simple graphs.

## 2. AN OVERVIEW OF THE MAIN ALGORITHM

The main algorithm proceeds as follows. We formulate a configuration integer program that contains, for each matching $M$ and time $t$, a variable $x_{Mt}$ that represents whether $M$ is to be scheduled at time $t$. We solve the linear programming relaxation by means of the ellipsoid algorithm. We then randomly round the variables in a way to be explained shortly. This results in a many-to-many correspondence of matchings to time slots. This is resolved by first randomly reordering matchings assigned to the same time slot, and then scheduling the matching in the order of the rounding and the reordering. Finally, the edges left unscheduled after this phase are then scheduled by a simple greedy procedure that iteratively schedules the edges in the earliest possible time slot.

There are two twists to the randomized rounding. One is the introduction of a multiplier $\alpha$, eventually chosen to be 0.9076. Namely, the probability that matching $M$ is selected at time $t$ is $\alpha x_{Mt}$. This introduces a balance between the costs incurred by the edges of rounded matchings and edges scheduled in the greedy phase.

The other variation is that each $x_{Mt}$ value is chopped up into many tiny fractions that are rounded independently. This ensures a much stronger concentration of the combined rounding, which is crucial in the greedy phase. The main reason for using this “chopping” operation is purely technical: we want the expected number of edges incident to a node $v$ of degree $d_v$ that are not covered during the first phase to be roughly $e^{-\alpha} d_v$ even for vertices of small degree. See Proposition 3.5. The minimum number of chopping required is to $n^4$ pieces (as the analysis indicates). Smaller chopping will not carry the inequalities through and larger chopping will only worsen the approximation ratio.

## 3. AN APPROXIMATION ALGORITHM USING CONFIGURATION LP

Let $n$ be the number of vertices in the graph and $\Delta$ be the maximum degree. Let $\mathcal{M}$ denote the collection of all (possibly non-maximal) matchings in the graph, including the empty matching. For an edge $h$, let $\mathcal{M}_h$ denote the set of matchings that contain $h$. We form an $LP$ with a variable $x_{Mt}$ for each matching $M$ and each time slot $t$. Observe that the number of rounds in any reasonable schedule is at most $\Psi = 2\Delta - 1$, since each edge has at most $2\Delta - 2$ adjacent edges.

$$\text{Minimize} \qquad opt^* = \sum_{t,M} t|M| x_{Mt}$$

$$\text{subject to:} \qquad \sum_{M \in \mathcal{M}} x_{Mt} = 1 \qquad \text{for each } t \tag{1}$$

$$\sum_{M \in \mathcal{M}_h} \sum_{t} x_{Mt} = 1 \qquad \text{for each edge } h \tag{2}$$

$$x_{Mt} \geq 0 \qquad \text{for each } t \text{ and } M$$

Equality (1) ensures that in an integral solution each round contains exactly one matching (if a round is not used by the optimum then it is assigned an empty matching). Equality (2) indicates that each edge belongs to exactly one scheduled matching. Since every valid solution must obey these constraints, the linear programming relaxation is indeed a relaxation of our problem.

Proposition 3.1. *The LP can be solved exactly in polynomial time and we may assume the solution vector is a basic feasible solution.*

Proof. As the $LP$ has an exponential number of variables, we consider the dual program. Equality (1) receives a variable $y_t$ and Equality (2) receives a variable $z_h$. The dual LP is then:

$$\text{maximize} \qquad \sum_h z_h + \sum_t y_t \tag{3}$$

$$\text{subject to:} \qquad \sum_{h \in M} z_h + y_t \leq t \cdot |M| \qquad \text{for each round } t \text{ and matching } M \tag{4}$$

In spite of having an exponential number of inequalities, the dual LP can be approximated to an arbitrary precision by the ellipsoid algorithm when given a polynomial-time separation oracle. Such an oracle must answer, within arbitrary precision, whether a candidate fractional solution is $\{y_t\} \cup \{z_h\}$ is feasible, and in case it is not, output a violated inequality (see [Khachiyan 1980]). A violated inequality implies that for some $M$ and $t$:

$$\sum_{h \in M} z_h + y_t > t|M|.$$

To find a violated inequality we look for every $t$ for the largest weighted matching with weights $z_h - t$. The value of the solution is $\sum_{h \in M} z_h - t|M|$ for some matching $M$. Since we are maximizing $\sum_{h \in M} z_h - t|M|$, if $\sum_{h \in M} z_h - t|M| \leq y_t$, this inequality is true to every $M$. Otherwise, we found a violated constraint. As long as we find separating hyperplanes we continue doing so until a feasible solution is found (as one clearly exists). We may restrict the variables in the primal program to those that correspond to violated dual constraints found (as these dual constraint suffice to define the resulting dual solution). Thus, the number of variables is polynomial and the primal program can be solved within arbitrary precision in polynomial time.

We may also assume that the solution is a basic feasible solution (see for example [Jain 2001]). □

### 3.1 The randomized rounding algorithm

For simplicity of the presentation the algorithm and analysis are for simple graphs. We point out later the changes needed (in only one proof) if the graph has parallel edges. Let $\{x_{Mt}\}$ be the optimum fractional solution for the instance at hand. We assume that $\{x_{Mt}\}$ is a basic feasible solution. This means that the number of $x_{Mt}$ variables with positive values is at most the number of variables in the dual $LP$; this is at most the number of rounds plus the number of edges, or at most $2n^2$.

Let $1/2 \leq \alpha \leq 1$ be a universal constant whose value will be optimized later. For technical reasons (to be clarified later) we need to do a *chopping operation* described as follows. Each $x_{Mt}$ is replaced by a set of $n^4$ smaller fractions $y^1_{Mt}, \ldots, y^{n^4}_{Mt}$, each with a value of $x_{Mt}/n^4$. We think of $y^j_{Mt}$ as associated with a matching $M_{j,t} = M$ and "responsible" for an $x_{Mt}/n^4$ fraction of $x_{Mt}$.

The algorithm has a randomized phase followed by a deterministic phase. The random phase first assigns to each time slot $t$ a *bucket* $B_t$ of matchings, all provisionally requesting to occupy that slot. Since bucket $B_t$ may currently contain more than one matching, a procedure is needed to decide which unique matching to use in each round $t$. In this *spreading* procedure, the matchings are evenly distributed so that each *actual round* $t$ receives at most one matching, thus forming a proper schedule. Because of the spreading, the actual unique matching assigned to round $t$ may not even belong to $B_t$.

Formally, the algorithm is defined as follows:

(i) **The initial ordering step:**
We choose an arbitrary *initial order* on all the chopped $M_{j,t}$ matchings. The order is deterministic and obeys time slots in that if $t < t'$ then any $M_{j,t}$ must come in the initial order *before* $M'_{j',t'}$ regardless of $j', j, M', M$.

(ii) **The randomized rounding step:** For each $M'_{j,t}$, $1 \leq t \leq \Psi$ and $1 \leq j \leq n^4$, place $M'$ independently at random in bucket $B_{t'}$ with probability $\alpha \cdot y^j_{t',M'}$. Several matchings may be selected into the same bucket $B_{t'}$, indicating their provisional wish to be scheduled at time $t'$.

(iii) **The random reordering step:** Randomly reorder all matchings that reached bucket $B_t$.
**Remark:** The random ordering step is completely independent of the randomized rounding step.

(iv) **The spreading step:** Spread the resulting matchings, in order of bucket number and the chosen order within each bucket. Thus, all matchings of bucket $B_t$ will be scheduled before all matchings of bucket $B_{t+1}$, etc. For each $i$, let $m_i$ denote the number of matchings in $B_i$. By increasing bucket numbers $i$, the matchings in $B_i$ are assigned to rounds $\sum_{j=1}^{i-1} m_j + 1$ through $\sum_{j=1}^{i-1} m_j + m_i$, with the ordering of the $m_i$ matchings that belong to $B_i$ following the random order of the previous step.

(v) **The assignment step**: We assign each edge $h$ to the first matching $M$ that contains $h$ (in the order of spreading). If $M$ is the i-th matching in the order then the *finish time* of $h$ is $f(h) = i$. We then remove copies of $h$ from all other matchings.

Denote by $\mathcal{M}_1, \mathcal{M}_2, \ldots \mathcal{M}_\gamma$ the matchings chosen in this first phase. Let $E_c = \bigcup_{i=1}^{\gamma} \mathcal{M}_i$ be the set of edges covered in this first phase, and $E_u = E - E_c$ be the set of *uncovered* edges.

In the second phase, the edges of $E_u$ are now scheduled in an arbitrary sequence by a straightforward greedy procedure. Namely, add edges one by one to the current schedule (collection of matchings) as early as possible, i.e. if an edge $(z, v) \in E_u$ is scheduled at time $t$ then for every time step $t' < t$ either edge incident to node $z$ or an edge incident to node $v$ is scheduled at time $t'$. Let the final schedule be $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_\Phi\}$ where $\Phi \geq \gamma$.

### 3.2 Analysis

Recall that for each matching $M$ and each bucket $t$, there are $n^4$ tries of adding $M$ into $t$. Let $A^j_{Mt}$ be the event the $j$-th try succeeds and $Y^j_{Mt}$ the indicator random variable of $A^j_{Mt}$. Let $X_{Mt} = \sum_j Y^j_{Mt}$. Since $\Pr[A^j_{Mt}] = \alpha y^j_{Mt}$, by linearity of expectation, $\mathbf{E}[X_{Mt}] = \alpha \cdot x_{Mt}$.

Let $\mathcal{A}_h = \{A^j_{Mt} | j = 1, \ldots, n^4, M \in \mathcal{M}_h, t = 1, \ldots, \Psi\}$ be the set of events involving edge $h$. We order these events for $h$ according to the initial order (see the first step of the algorithm), and denote $\mathcal{A}_h = \{A^h_1, A^h_2, \ldots, A^h_{k_h}\}$ according to this order. The $i$-th event $A^h_i$ in this order corresponds to some $A^j_{M',t'}$ that is the $j$th trial of some pair $t', M'$. Correspondingly, let $y^h_i$ denote the chopped value $y^j_{M',t'}$, namely, $y^h_i = \Pr[A^j_{t'M'}]/\alpha$. Note that the order above depends for now only on the initial order and does not depend on the later random ordering in buckets. Let $k_h$ be the total number of events related to $h$.

Let $deg(v)$ be the degree of vertex $v$, and $d_c(v)$ be the number of edges incident on $v$ in $E_c$ (as a random variable).

Let the *finish time* of an edge $h$, denoted $f(h)$, be the round $i$ in which $h$ was scheduled, i.e., if the unique matching containing $h$ was $\mathcal{M}_i$. We split this quantity into two parts: its *covering contribution* $f_c(h)$ and its *uncovered contribution* $f_u(h)$. If $h \in E_c$, then $f_c(h) = f(h)$ and $f_u(h) = 0$, while if $h \in E_u$, then $f_c(h) = 0$ and $f_u(h) = f(h)$. Clearly, $f_c(h) + f_u(h) = f(h)$. Thus, $f_c(h)$ ($f_u(h)$) correspond to the amount contributed by $h$ to the objective function from the first (second) phase, respectively.

**Remark:** Throughout, we assume, without explicit mention, that $n$ is large enough (larger than any universal constant required in the analysis).

3.2.0.2 *Outline of the proof.* We analyze the performance of the algorithm by bounding individually the expected covering and uncovered contribution of each edge. See Figure 1 for an overview of the structure of the proof.

The unconditional covering contribution is obtained in Lemma 3.7. For this, we use the initial ordering defined on the events involving the edge $h$ and first evaluate the expected covering contribution conditioned on the edge being selected due to the first event and more generally $i^{\text{th}}$ event in the ordering, respectively (Proposition 3.2). We use several propositions to manipulate these conditional contributions: 3.3, 3.8, 3.9. Additionally, it depends on Proposition 3.5 which bounds the probability that an edge remains uncovered after the randomized phase.

To bound the uncovered contribution, we charge each edge $h$ by some quantity $f'_u(h)$ (see Definition 3.11) so that $\sum_{h \in E_u} f'_u(h) = \sum_{h \in E_u} f_u(h)$. This is done because analyzing the expectation of $f'_u(h)$ is easier. We bound the expected number of covered edges incident on a vertex (Proposition 3.10) and then argue the uncovered contribution of each edge (Lemma 3.12) using $f'_u(h)$. Finally, summing over all the edges of the graph, we combine the two contributions to derive the approximation in Theorem 3.15. We note that the part of the ratio corresponding to the covering contributions is in terms of the LP objective, while the uncovered contributions is in terms of a weaker previously studied lower bound on the optimal value (Definition 3.13).

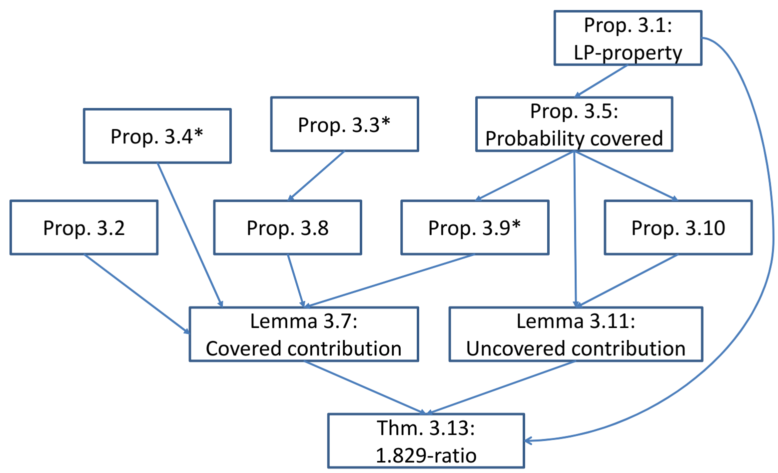


Fig. 1. Structure of the proof of the main result. Asterisk denotes that proof is given in the appendix.

3.2.0.3 *Proof details.* We first bound $f_c(h)$, conditioned on $h \in E_c$. Recall that $\mathcal{A}_h = \{A_1^h, A_2^h, \ldots A_{k_h}^h\}$ is the initial ordering of matchings containing $h$. At least one of the events in this list has succeeded.

We compute the finish time of $h$ under the assumption that events $A_1^h, \ldots, A_p^h$ failed and the first to succeed was $A_{p+1}^h$, for some $p \geq 0$. Let $t_i^h$ be the slot number in the initial ordering of the event $A_i^h$.

PROPOSITION 3.2. *The expected covering contribution of edge $h$, given that the $p+1$-th event in $\mathcal{A}_h$ was the first one to occur, is bounded by*

$$\mathbf{E}\left[ f_c(h) \,\middle|\, \bigcap_{i \leq p} \overline{A_i^h} \cap A_{p+1}^h \right] \leq \alpha \cdot \left( t_{p+1}^h - \frac{1}{2} \sum_{i=1}^{p+1} y_i^h + \frac{1}{\alpha} - \frac{1}{2} \right) .$$

PROOF. We bound the expected waiting time of an edge, or the expected number of matchings that precede its round. We first bound the number of those coming from earlier buckets, and then those coming from the same bucket as $h$.

The expected sum of fractions chosen per bucket is $\alpha \cdot \sum_{M \in \mathcal{M}} x_{Mt} = \alpha$. Thus the unconditional expected waiting time of $h$ due to the $t_{p+1} - 1$ first rounds is $\alpha(t_{p+1}^h - 1)$. However, we are given that previous events involving $h$ did not occur. Let $B = \{A_i^h,\ i \leq p \mid t_i^h < t_{p+1}^h\}$ be the set of events concerning $h$ that belong to earlier buckets. Then, $h$ waits

$$\alpha \left( t_{p+1}^h - 1 - \sum_{A_i^h \in B} y_i^h \right) \tag{5}$$

in expectation for previous buckets.

We now consider the matchings belonging to the current bucket $t_{p+1}^h$. Let $W = \{A_i^h \mid i \leq p, t_i^h = t_{p+1}^h\}$ be the set of events involving $h$ that precede it in the initial order but also concern the same bucket. The expected number of matchings in bucket $t^h_{p+1}$, conditioned on $\cap_{A\in W}\overline{A}$ (i.e. none of its preceding events involving $h$ occurring), equals

$$\alpha\left(1-\sum_{A^h_i\in W} y^h_i\right) .$$

This amount is independent of the random ordering step, but the matchings will be spread within the bucket randomly. Hence, taking the expectation also over the random orderings, the waiting time of $h$ for this bucket is at most

$$\alpha\left(1/2-\sum_{A^h_i\in W} y^h_i/2 - y^h_{p+1}/2\right) . \tag{6}$$

We now add the waiting times of (5) and (6), observing that worst case occurs when $B=\emptyset$. In that case the expected waiting time is bounded by

$$\alpha\left(t^h_{p+1}-\frac{1}{2}-\sum_{i=1}^{p+1} y^i_h/2\right) .$$

Adding the round of $M^h_{p+1}$ itself yields the claim. □

**Remark:** The above number can indeed be strictly larger then the round of $h$. The round of $h$ can be smaller if the following events happen. There is a matching containing $h$ located *after* $M^h_{p+1}$ in the initial ordering, this matching reaches slot $t^h_{p+1}$ and is located before $M^h_{p+1}$ by the random ordering. However, it seems hard to use this fact to improve the ratio.

The proof of the next claim appears in the appendix

PROPOSITION 3.3. *Let $y_1, y_2, \ldots, y_k$ be non-negative numbers satisfying $\sum_{i=1}^k y_i = 1$. Then,*

$$\frac{y_1^2}{2} + y_2\cdot\left(y_1+\frac{y_2}{2}\right) + y_3\left(y_1+y_2+\frac{y_3}{2}\right) + \ldots + y_k\cdot\sum_{i=1}^{k-1} y_i + \frac{y_k^2}{2} = \frac{1}{2} . \tag{7}$$

The next claim appeared many times in the approximation algorithms literature. Chudak and Shmoys [2003] were the first ones to use it. A similar proof appears in [Sviridenko 2002].

PROPOSITION 3.4. *Let $t_1, \ldots, t_k$ be positive numbers so that $t_1 \le t_2 \le \cdots \le t_k$, and $x_1, \ldots, x_k$ be positive numbers such that $\sum_{i=1}^k x_i \le 1$. Then:*

$$t_1\cdot x_1 + t_2\cdot(1-x_1)\cdot x_2 + \cdots + \prod_{i=1}^{k-1}(1-x_i)\cdot x_k t_k \le \frac{\left(1-\prod_{i=1}^k(1-x_i)\right)\cdot\sum_{i=1}^k t_i x_i}{\sum_{i=1}^k x_i} .$$

PROPOSITION 3.5. *The probability that an edge $h$ is not covered is bounded by*

$$\frac{1}{e^\alpha}\left(1-\frac{3}{n^2}\right) \le \Pr[h\in E_u] = \prod_{i=1}^{k_h}\left(1-\alpha y^h_i\right) \le \frac{1}{e^\alpha} .$$

Proof. Observe that

$$\sum_{i=1}^{k_h} y_i^h = \sum_t \sum_{M\in\mathcal{M}_h} \sum_{j=1}^{n^4} y_{Mt}^j = \sum_t \sum_{M\in\mathcal{M}_h} x_{Mt} = 1, \tag{8}$$

by constraint (2) of the LP. Therefore, the upper bound follows from the fact that $1 - x \le e^{-x}$.

To get the lower bound we need to group the $y_{Mt}^j$ variables according to their $M, t$ values:

$$\Pr[h \in E_u] = \prod_{t,M\in\mathcal{M}_h} \prod_{i=1}^{n^4} (1 - \alpha \cdot y_{Mt}^i) = \prod_{t,M\in\mathcal{M}_h} \left(1 - \frac{\alpha \cdot x_{Mt}}{n^4}\right)^{n^4}.$$

The following inequality holds:

$$\left(1 - \frac{\alpha \cdot x_{Mt}}{n^4}\right)^{n^4} \ge \frac{1}{e^{\alpha x_{Mt}}} \cdot \left(1 - \frac{\alpha \cdot x_{Mt}}{n^4}\right)^{\alpha \cdot x_{Mt}}$$

because $(1 - 1/x)^{x-1} > 1/e$ holds for all $x > 1$. As $x_{Mt} \le 1$ and $\alpha \le 1$ we get:

$$\left(1 - \frac{\alpha \cdot x_{Mt}}{n^4}\right)^{n^4} \ge \frac{1}{e^{\alpha x_{Mt}}} \cdot \left(1 - \frac{1}{n^4}\right).$$

By assumption, $\{x_{Mt}\}$ is a basic feasible solution. Hence, the number of non-zero $x_{Mt}$ is at most the number of edges plus the number of rounds, or at most $2n^2$. Thus

$$\prod_{i=1}^{k_h} (1 - y_i^h) \ge \left(1 - \frac{1}{n^4}\right)^{2n^2} \cdot \prod_{t,M\in\mathcal{M}_h} \frac{1}{e^{\alpha x_{Mt}}} \ge \frac{1}{e^{\alpha}} \left(1 - \frac{3}{n^2}\right).$$

We used Equality (2) in the LP and $(1 - 1/n^4)^{2n^2} \ge (1 - 3/n^2)$. □

**Remark:** More generally, if there are parallel edges, the fractions need to be chopped into at least $|E| \cdot n^2$ pieces.

*Notation* 3.6. Let $b_i^h = \alpha \cdot \left( t_i^h - \left( \sum_{j=1}^{i} y_j^h \right) /2 + 1/\alpha - 1/2 \right)$.

Lemma 3.7. *The expected covering contribution of an edge $h$ is bounded by*

$$\mathbf{E}\left[f_c(h)\right] \le \left(1 + \frac{1}{n}\right) \cdot \left(1 - \frac{1}{e^{\alpha}}\right) \cdot \left( \left(1 - \frac{3\alpha}{4}\right) + \alpha \sum_{t,M\in\mathcal{M}_h} x_{Mt} \cdot t \right).$$

PROOF. By Proposition 3.2, $\mathbf{E}\left[f_c(h) \mid \bigcap_{j=1}^{i-1} \overline{A_j^h} \cap A_i^h\right] \leq b_i^h$. Thus,

$$\begin{aligned}
\mathbf{E}\left[f_c(h)\right] &= \mathbf{E}\left[f_c(h) \mid A_1^h\right] \cdot Pr[A_1^h] + \mathbf{E}\left[f_c(h) \mid \overline{A_1^h} \cap A_2^h\right] \cdot \Pr[\overline{A_1^h} \cap A_2^h] + \ldots \\
&\quad + \mathbf{E}\left[f_c(h) \mid \bigcap_{i<k_h} \overline{A_i^h} \cap A_{k_h}^h\right] \cdot \Pr[\cap_{i<k_h} \overline{A_i^h} \cap A_{k_h}^h] + 0 \cdot \Pr[\bigcap_{i\leq k_h} \overline{A_i^h}] \\
&\leq \alpha \cdot y_1^h \cdot b_1^h + \left(1-\alpha\cdot y_1^h\right)\cdot \alpha \cdot y_2^h \cdot b_2^h + \left(1-\alpha\cdot y_1^h\right)\cdot\left(1-\alpha\cdot y_2^h\right)\cdot \alpha y_3^h \cdot b_3^h \\
&\quad + \ldots + \prod_{i=1}^{k_h-1}\left(1-\alpha\cdot y_i^h\right)\alpha\cdot y_{k_h}^h \cdot b_{k_h}^h \\
&\leq \left(1-\left(\prod_{i=1}^{k_h}\left(1-\alpha\cdot y_i^h\right)\right)\right)\cdot \sum_{i=1}^{k_h} y_i^h b_i^h \;. \qquad \text{(By Proposition 3.4)}
\end{aligned}$$

The $\alpha$ term that multiplied every term before the last inequality was canceled because $\sum_{i=1}^{k_h} \alpha y_i^h = \alpha$.

The lemma now follows from the combination of the following two propositions. □

PROPOSITION 3.8. $\displaystyle\sum_{i=1}^{k_h} y_i^h b_i^h \leq \alpha \sum_{t,M\in\mathcal{M}} x_{Mt}\cdot t + (1-3\alpha/4)$.

PROOF. Recall that

$$y_i^h b_i^h = \alpha y_i^h t_i^h + \alpha \cdot \left(y_i^h \cdot \left(-\frac{1}{2}\sum_{j=1}^{i} y_j^h + \frac{1}{\alpha} - \frac{1}{2}\right)\right) \;.$$

We break the sum into three parts, and analyze first the total contribution of the two last terms $(-\alpha \cdot y_i^h \left(\sum_{j=1}^{i} y_j^h\right)/2$ and $\alpha \cdot y_i^h(1/\alpha - 1/2))$ when summing over all $i$. By Proposition 3.3,

$$-\alpha \sum_{i=1}^{k_h} y_i^h \cdot \left(\frac{1}{2}\sum_{j=1}^{i} y_j^h\right) \leq -\frac{\alpha}{2}\sum_{i=1}^{k_h} y_i^h \cdot \left(\sum_{j=1}^{i-1} y_j^h + \frac{y_i^h}{2}\right) = -\alpha/4 \;. \tag{9}$$

Since $\sum_{i=1}^{k_h} y_i^h = 1$ we have that

$$\alpha\left(\frac{1}{\alpha}-\frac{1}{2}\right)\cdot\sum_{i=1}^{k_h} y_i^h = \left(1-\frac{\alpha}{2}\right) \;. \tag{10}$$

Finally, consider the sum of the terms $\alpha y_i^h t_i^h$. By re-indexing, we have that

$$\sum_{i=1}^{k_h} \alpha y_i^h \cdot t_i^h = \alpha \sum_{t,M\in\mathcal{M}_h} \sum_{i=1}^{n^4} y_{Mt}^i \cdot t = \alpha \sum_{t,M\in\mathcal{M}_h} x_{Mt}\cdot t \;. \tag{11}$$

Adding (9), (10) and (11) now yields the claim. □

The proof of the following lemma is given in the appendix.

Proposition 3.9. $\left(1-\left(\prod_{i=1}^{k_h}\left(1-\alpha\cdot y_i^h\right)\right)\right)\leq\left(1+\frac{1}{n}\right)\cdot\left(1-\frac{1}{e^\alpha}\right)$ .

We now turn to bounding the expected contribution of uncovered edges.

Proposition 3.10. *The expected number of covered edges incident on a vertex $v$ is bounded by*

$$\mathbf{E}\left[d_c(v)\right]\leq\left(1+\frac{1}{n}\right)\left(1-\frac{1}{e^\alpha}\right)deg(v)\ .$$

Proof. By Proposition 3.5, we have for each $h\in E(v)$ that

$$\Pr[h\in E_c]=1-\Pr[h\in E_u]\leq 1-\frac{1}{e^\alpha}\left(1-\frac{3}{n^2}\right)=\left(1-\frac{1}{e^\alpha}\right)+\frac{3}{n^2\cdot e^\alpha}.$$

Using $\alpha\geq 1/2$, we have that, for large enough $n$,

$$\Pr[h\in E_c]\leq\left(1+\frac{1}{n}\right)\left(1-\frac{1}{e^\alpha}\right)\ .$$

By linearity of expectation,

$$\mathbf{E}\left[d_c(v)\right]\leq\left(1+\frac{1}{n}\right)\left(1-\frac{1}{e^\alpha}\right)deg(v)\ .$$

□

**Remark:** It appears difficult to bound the expectation of $d_c(v)$ as above without the chopping operation, especially when degrees are small.

We are ready to bound from above the expectation of the sum of finish times in the greedy phase. Consider an uncovered edge $(u,v)\in E_u$. This edge would first have to wait a total of at most $d_c(v)+d_c(u)$ rounds until all covered edges incident on $v$ and $u$ are scheduled. After that, each time the edge $(u,v)$ is not selected, at least one of $u$ or $v$ is matched. Thus, each time the edge $(u,v)$ waits can be charged to an edge in $E_u$ that is incident on either $u$ or $v$. Thus, the contribution of the greedy step to the sum of finish times is at most

$$\sum_{v\in V}\sum_{i=d_c(v)+1}^{deg(v)} i=\sum_{v\in V}\left(\sum_{i=1}^{deg(v)} i-\sum_{i=1}^{d_c(v)} i\right)=\sum_{w\in V}\left(\binom{deg(w)+1}{2}-\binom{d_c(w)+1}{2}\right)\ . \tag{12}$$

Now, divide the term $\binom{deg(v)+1}{2}-\binom{d_c(v)+1}{2}$, equally among the edges in $E_u(v)$. Then, the edge $h=(z,v)$ is charged

$$\begin{aligned}&\frac{\binom{deg(z)+1}{2}-\binom{d_c(z)+1}{2}}{deg(z)-d_c(z)}+\frac{\binom{deg(v)+1}{2}-\binom{d_c(v)+1}{2}}{deg(v)-d_c(v)}\\&=\frac{deg(z)+d_c(z)}{2}+\frac{deg(v)+d_c(v)}{2}+1\ .\end{aligned}$$

*Definition* 3.11. Define

$$f'_u(h)=\begin{cases}\frac{deg(z)+d_c(z)}{2}+\frac{deg(v)+d_c(v)}{2}+1, & \text{if } h\in E_u\\ 0, & \text{otherwise.}\end{cases}$$

Observe that $\sum_{h\in E} f'_u(h) \geq \sum_{h\in E} f_u(h)$. Hence, for the purpose of evaluating the sum of the expected values, it is enough to bound the expectation of $f'_u(h)$.

LEMMA 3.12. *The contribution of an uncovered edge $h = (z, v)$ is bounded by*

$$\mathbf{E}\left[f'_u(h)\right] \leq \left(1+\frac{1}{n}\right)\cdot\frac{1}{e^{\alpha}}\left[\left(2-\frac{1}{e^{\alpha}}\right)\cdot\left(\frac{deg(z)+deg(v)}{2}+1\right)-\left(1-\frac{1}{e^{\alpha}}\right)\right]$$

PROOF. The probability that $h$ is uncovered is at most $1/e^{\alpha}$ (see Proposition 3.5). Thus,

$$\begin{aligned}\mathbf{E}\left[f'_u(h)\right] &\leq \frac{1}{e^{\alpha}}\cdot\mathbf{E}\left[\frac{deg(z)+d_c(z)}{2}+\frac{deg(v)+d_c(v)}{2}+1\right]\\ &\leq \frac{1}{e^{\alpha}}\left(1+\frac{1}{n}\right)\left(2-\frac{1}{e^{\alpha}}\right)\left(\frac{deg(z)+deg(v)}{2}+1\right)\\ &\quad -\frac{1}{e^{\alpha}}\left(1+\frac{1}{n}\right)\left(1-\frac{1}{e^{\alpha}}\right). \qquad \text{(By Proposition 3.10)}\end{aligned}$$

□

The following measure has been useful for lower bounding the cost of the optimal solution.

*Definition* 3.13. Let $q(G) = \sum_{v\in V}\binom{deg(v)+1}{2}$.

Let $opt(G)$ denote the cost of the optimal sum edge coloring of $G$, and recall that $opt^*$ denotes the value of the linear program. It was shown in [Bar-Noy et al. 1998] that $opt(G) \geq q(G)/2$.

*Observation* 3.14. $opt(G) \geq q(G)/2 = \sum_{(z',v')\in E}\left(\frac{deg(z')+deg(v')}{4}+\frac{1}{2}\right)$

PROOF.

$$\begin{aligned}q(G)/2 &= \sum_{v\in V}\frac{deg(v)(deg(v)+1)}{4} = \sum_{v\in V}\frac{deg^2(v)}{4}+\frac{|E|}{2}\\ &= \sum_{(z',v')\in E}\left(\frac{deg(z')+deg(v')}{4}+\frac{1}{2}\right).\end{aligned}$$

□

THEOREM 3.15. *The expected approximation ratio of the algorithm is at most 1.8298.*

PROOF. Each edge $h'$ contributes two terms that depend *only on $n$ and $\alpha$*. There is the positive contribution (see Lemma 3.7) of

$$\left(1+\frac{1}{n}\right)\cdot\left(1-\frac{1}{e^{\alpha}}\right)\cdot\left(1-\frac{3\alpha}{4}\right) \tag{13}$$

and the negative term from Lemma 3.12 of

$$-\frac{1}{e^{\alpha}}\cdot\left(1+\frac{1}{n}\right)\cdot\left(1-\frac{1}{e^{\alpha}}\right). \tag{14}$$

This amounts to

$$\left(1+\frac{1}{n}\right)\cdot\left(1-\frac{1}{e^{\alpha}}\right)\cdot\left(1-\frac{3\alpha}{4}-\frac{1}{e^{\alpha}}\right) . \tag{15}$$

For $\alpha \geq 0.606$, the above term is negative and only makes the upper bound smaller.

Ignoring the terms (13) and (14) we get:

$$\begin{aligned}
E\left[\sum_h f(h)\right] &= \sum_h \left(\mathbf{E}\left[f_c(h)\right]+\mathbf{E}\left[f_u(h)\right]\right) = \sum_h \left(\mathbf{E}\left[f_c(h)\right]+\mathbf{E}\left[f'_u(h)\right]\right)\\
&\leq \sum_h \left(\left(1+\frac{1}{n}\right)\alpha\cdot\left(1-\frac{1}{e^{\alpha}}\right)\sum_{t,M\in\mathcal{M}_h} x_{Mt}\cdot t\right)\\
&\quad + \sum_{h=(z,v)}\left(\left(1+\frac{1}{n}\right)\frac{2}{e^{\alpha}}\left(2-\frac{1}{e^{\alpha}}\right)\cdot\left(\frac{deg(z)+deg(v)}{4}+\frac{1}{2}\right)\right)\\
&= \left(1+\frac{1}{n}\right)\alpha\left(1-\frac{1}{e^{\alpha}}\right)opt^* + \left(1+\frac{1}{n}\right)\frac{2}{e^{\alpha}}\left(2-\frac{1}{e^{\alpha}}\right)\cdot\frac{q(G)}{2}\\
&\leq \left(1+\frac{1}{n}\right)\cdot\left(\alpha\cdot\left(1-\frac{1}{e^{\alpha}}\right)+\frac{2}{e^{\alpha}}\left(2-\frac{1}{e^{\alpha}}\right)\right)opt(G) ,
\end{aligned}$$

where we used Lemmas 3.7 and 3.12 in the first inequality and Observation 3.14 in the second inequality.

For $\alpha = 0.9076$ and large enough $n$, the right hand side evaluates to $1.8298 opt(G)$. □

## 4. COMBINATORIAL APPROACH

We give a combinatorial algorithm that applies to simple graphs. We use the algorithm ACS of [Halldórsson et al. 2003]. This algorithm is designed to operate in rounds, finding in each round a sub-solution of maximum throughput.

A *b-matching* is a subgraph where each node has at most $b$ incident edges. A maximum weight $b$-matching can be found in polynomial time by a reduction to matching (cf. [Cook et al. 1998]). Denote this algorithm by $b$-Matching.

In each round $i$, the algorithm ACS finds a maximum $k_i$-matching, where $k_1, k_2, \ldots$ forms a geometric sequence. The base $q$ of the sequence is a parameter to the algorithm, while the offset $\alpha$ is selected uniformly at random from the range $[0, 1)$. We show later how to derandomize this strategy. The $b$-matching is then turned into a collection of matchings, using Vizing's algorithm, and those are then scheduled in the natural non-decreasing order of size. See Figure 2.

This algorithm attains a performance ratio of 1.796 when an algorithm is available for obtaining an maximum $k$-(edge)-colorable subgraph [Halldórsson et al. 2003]. This leads to a 1.796-approximation of BPSMS in bipartite graphs [Gandhi et al. 2008], since in bipartite graphs a maximum $k$-matching is a maximum $k$-(edge)-colorable subgraph. The main issue for analysis is to assess the extra cost per round due to the application of Vizing's algorithm. We use ideas from [Epstein et al. 2008] for simplifying parts of the analysis.

4.0.0.4 *Analysis.* The analysis proceeds as follows. After defining a simple lower bound on the cost of edges in an optimal solution, we introduce a series of notation,

```
ACS(G,q)
  α = U[0,1); i ← 0
  while (G ≠ ∅) do
    k_i = ⌊q^(i+α)⌋
    G_i ← b-Matching(G,k_i).
    Color the edges of G_i using Vizing's algorithm
    Schedule the colors in non-increasing order of cardinality
    G ← G − G_i
    i ← i + 1;
end
```

Fig. 2. Combinatorial algorithm ACS for simple graphs

followed by an expression that we show forms an upper bound on the amortized cost of each edge in the algorithm's solution. We simplify it and break it into three parts: the asymptotic cost, the logarithmic charge for the application of Vizing's algorithm, and additive incidental cost. The main effort lies in evaluating the last one precisely. In particular, we utilize that when the first block involves a 1-matching, then we already have a single matching and avoid paying the additive 1 incurred by Vizing's algorithm. The performance ratio is then derived from these upper and lower bounds.

For any natural number $r$, denote by $d_r$ the minimum number of colors needed to color at least $r$ edges from $E$. Observe the following easy bound.

*Observation* 4.1. $\sum_{r=1}^{n} d_r \leq opt(G)$.

Let the edges be indexed $r = 1, 2, \ldots, m$ in agreement with the schedule formed by ACS. Let the *block* containing edge $r$ denote the iteration $i$ of ACS (starting at 0) in which the edge is scheduled. Let $\delta$ be the characteristic function of the event that the first block is of size 1, i.e. $\delta = 1$ if $\alpha < \log_q 2$ and 0 otherwise. Let $b_r$ be the minimum value $i$ such that $d_r \leq k_i$. Let $\beta_r$ be the characteristic function of the event that $b_r$ is positive, i.e. $\beta_r = 1$ if $\alpha < \min(\log_q d_r, 1)$ and 0 otherwise. We analyze the following amortized upper bound on the algorithm's cost of coloring edge $r$:

$$\pi_r = \sum_{i=0}^{b_r - 1} (k_i + 1) + \frac{k_{b_r} + 2}{2} - \frac{\beta_r + 1}{2} \delta \ .$$

Lemma 4.2. $ACS(G) \leq \sum_{r=1}^{m} \pi_r$.

Proof. Define $b'_r$ as the block number of edge $r$. Recall that a maximum $b$-matching contains at least as many edges as can be colored with $b$ colors. Thus, $b'_r \leq b_r$, for all $r$. We first argue that $ACS(G) \leq \sum_{r=1}^{m} \pi'_r$, where

$$\pi'_r = \begin{cases} \sum_{i=0}^{b'_r - 1} (k_i + 1) + \frac{k_{b'_r} + 2}{2} - \delta & \text{if } b_{r'} > 0 \\ \frac{k_0 + 2 - \delta}{2} & \text{if } b_{r'} = 0. \end{cases} \tag{16}$$

First, let us consider edges $r$ that fall into block later than the first one, i.e. satisfy $b'_r > 0$. The first two terms of the first case of (16) bound the amortized cost of coloring the edge $r$, under the assumption that all blocks use an additional color due to the application of Vizing's theorem. The first term represents the number of colors used on blocks preceding that of $r$'s. Each block is formed by a $k_i$-matching and is colored into $k_i+1$ matchings. The second term represents an upper bound on the contribution of the last block. This bound is the mean over two orderings of the block: some arbitrary ordering and its reverse. The cost of the order obtained by the algorithm is at most the cost of the smaller of these two. The last term, $\delta$, takes into account the fact that the first block, when of unit size, has no additive cost. Thus, we subtract the additive 1 for the first block when the first block is of unit size, i.e. when $\alpha < \log_q 2$.

Consider now nodes $r$ that fall into the first block, i.e. $b'_r = 0$. The size of that block is $k_0 = \lfloor q^\alpha \rfloor$. If $\alpha < \log_q 2$, the block is of unit size and the node will be colored $\pi'_r = \frac{1+2-1}{2} = 1$. If $\alpha \geq \log_q 2$, the block is of non-unit size, and $k_0+1$ colors will be used. The average cost of nodes in the block is again at most the average of two opposite orderings, or $(k_0+2)/2 = \pi'_r$.

Finally, we claim that $\sum_{r=1}^m \pi'_r \leq \sum_{r=1}^m \pi_r$. Recall that $b'_r \leq b_r$. Observe that when $b'_r = b_r$, for all $r$, then $\pi'_r = \pi_r$. We argue that $\pi'_r$ is monotone with the values of $\{b'_r\}_r$. Namely, it is easy to verify that if one $b'_r$ value increases, the cost of some $\pi'_r$ values will strictly increase and none will decrease. Hence, the claim. $\square$

We split our treatment into three terms that intuitively represent the asymptotic contribution of the $r$-th edge, the block count, and the nearly-additive incidental cost. The following characterization is obtained by rewriting $k_0$ as $(\lfloor q^\alpha \rfloor - q^\alpha) + q^\alpha$.

Lemma 4.3. *For each edge $r$,*

$$\pi_r \leq \varphi_r \left[\frac{1}{q-1} + \frac{1}{2}\right] + b_r + 1 - \frac{q^\alpha}{q-1} + \gamma_r \ .$$

*where $\varphi_r = q^{b_r+\alpha}$ and*

$$\gamma_r = \frac{\beta_r + 1}{2} \left(\lfloor q^\alpha \rfloor - q^\alpha - \delta\right) \ .$$

We first bound the intermediate quantity $b_r$, roughly corresponding to the the number of the block containing the $r$-th edge.

Lemma 4.4. $\mathbf{E}[b_r] = \log_q d_r$.

Proof. Let $s = \log_q d_r - \alpha$. Thus, $d_r = q^{s+\alpha} = \lceil q^{s+\alpha} \rceil$. Recall that $b_r$ is the smallest integer $i$ such that $\lfloor q^{s+\alpha} \rfloor = d_r \leq \lfloor q^{i+\alpha} \rfloor$. Thus, $b_r = \lceil s \rceil$. Rewrite as $b_r = s + (\lceil s \rceil - s) = \log_q d_r + (\lceil s \rceil - s) - \alpha$. Since $\alpha \sim \mathbf{U}[0,1)$, so is $\lceil s \rceil - s$. Hence, $\mathbf{E}[b_r] = \log_q d_r + \mathbf{E}[\lceil s \rceil - s] - \mathbf{E}[\alpha] = \log_q d_r + 1/2 - 1/2 = \log_q d_r$. $\square$

The following bound on the asymptotic costs of $\pi_r$ was given in [Halldórsson et al. 2003] (see also [Epstein et al. 2008]).

Lemma 4.5. $\mathbf{E}[\varphi_r] = \frac{q-1}{\ln q} d_r$.

Proof. Recall from the previous lemma that $b_r = \lceil s \rceil$, where $s = \log_q d_r - \alpha$. Hence, $\varphi_r = q^{b_r - s} d_r = q^{\lceil s \rceil - s} d_r$. Since $\alpha \sim \mathbf{U}[0,1)$, so is $\lceil s \rceil - s$. Thus,

$$\frac{\mathbf{E}[\varphi_r]}{d_r} = \mathbf{E}\left[\frac{\varphi_r}{d_r}\right] = \mathbf{E}\left[q^{\lceil s \rceil - s}\right] = \mathbf{E}[q^\alpha] = \int_0^1 q^x \, dx = \frac{q-1}{\ln q} \ .$$

□

We then evaluate the incidental costs in the following lemma, whose rather lengthy proof is deferred to the appendix.

LEMMA 4.6. *For* $q \in [3,4)$,

$$\mathbf{E}[\gamma_r] = \begin{cases} \frac{1}{2}\left(3 - \log_q 12 - \frac{q-1}{\ln q}\right) & \text{if } d_r = 1 \\ \frac{1}{2}\left(3 - \log_q 12 - \frac{q}{\ln q}\right) & \text{if } d_r = 2 \\ \frac{1}{2}\left(3 - \log_q 16/3 - \frac{q+1}{\ln q}\right) & \text{if } d_r = 3, \text{ and} \\ 3 - \log_q 12 - \frac{q-1}{\ln q} & \text{if } d_r \geq 4. \end{cases}$$

We first argue a slightly weaker bound for the algorithm.

THEOREM 4.7. *The performance ratio of ACS is at most* $\max_r \frac{\mathbf{E}[\pi_r]}{d_r} \leq 1.90488$.

PROOF. By Lemma 4.2 and Observation 4.1 it suffices to show that $\mathbf{E}[\sum_{r=1}^m \pi_r] \leq 1.90488 \sum_{r=1}^m d_r$. We use $q = 3.666$, which is slightly larger than the value used in [Halldórsson et al. 2003] when there was no additive overhead.

By Lemmas 4.3, 4.4 and 4.5, we have for any $r$ that

$$\begin{aligned} \mathbf{E}[\pi_r] &\leq \frac{q-1}{\ln q} d_r \left[\frac{1}{q-1} + \frac{1}{2}\right] + \log_q d_r + 1 - \frac{1}{\ln q} + \gamma_r \\ &\leq 1.79586 d_r + \log_q d_r + 0.23042 + \gamma_r \ . \end{aligned}$$

Thus, using 4.6, we have that $\mathbf{E}[\pi_r] \leq \tau(d_r) \cdot d_r$, where

$$\tau(t) \doteq \begin{cases} 1.54361 & \text{if } t = 1 \\ 1.74407 & \text{if } t = 2 \\ 1.84111 & \text{if } t = 3, \text{ and} \\ 1.79586 + \frac{\log_q t - .73474}{t} & \text{if } t \geq 4. \end{cases} \tag{17}$$

The function $\tau$ is concave with integral-valued maximum of 1.90488 at $d = 7$ (while the real-valued function $f(x) = 1.79586 + \frac{\log_q x - .73474}{x}$ has maximum at $x = exp(0.769763 - 0.73474) \approx 7.06$). The function is plotted in Figure 3. This establishes that $\mathbf{E}[\pi_r] \leq 1.90488 d_r$, for any $r$, and thus the theorem. □

A slightly better performance ratio can be argued by considering the combined cost of groups of edges. Intuitively, the greedy nature of the ACS algorithm ensures that for each edge $r$ with $d_r = k$, there are corresponding edges $r'$ with all the $d$-values from 1 to $k-1$.

THEOREM 4.8. *The performance ratio of ACS is at most* $\max_t \frac{\sum_{d=1}^t d \cdot \tau(d)}{\binom{t+1}{2}} \leq 1.8886$.

PROOF. Consider a fixed optimal solution *opt*. Let $o_r$ be the color assigned to the $r$-th edge. Let $f(t) = |\{r \in E : o_r = t\}|$ be the number of edges with colored $t$ by *opt*, for $t \geq 1$, and let $g(y) = |\{x : f_x \geq y\}|$ be the number of color classes in *opt*

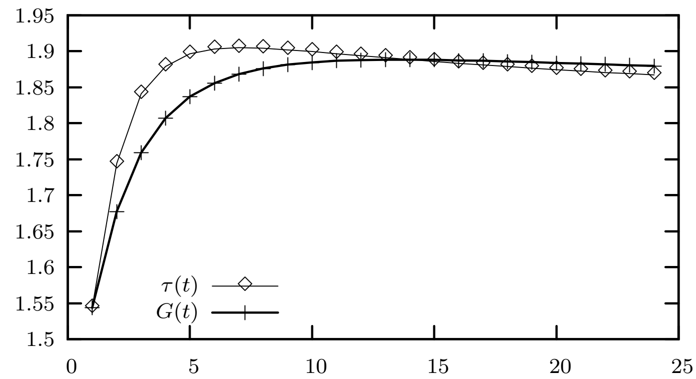


Fig. 3. Plots of the per-edge bound of $\tau(t)$ and the amortized bound of $G(t)$.

with $y$ or more edges, for $y \geq 1$. Observe that $f$ is monotone non-decreasing, and thus the color classes containing at least $y$ edges are precisely classes $1, 2, \ldots, g(y)$. We can write

$$opt(G) = \sum_r o_r = \sum_{t=1} f(t) \cdot t = \sum_{y=1} \binom{g(y)+1}{2} \ . \tag{18}$$

With $k$ colors, *opt* can color at most as many edges as are contained in a maximum $k$-edge-colorable subgraph. Thus, $o_r \leq d_r$, and $\mathbf{E}[ACS(G)] \leq \sum_r \tau(d_r) \cdot d_r \leq \sum_r \tau(o_r) \cdot o_r$. We can also rewrite this as

$$\mathbf{E}[ACS(G)] \leq \sum_{t=1} \tau(t) \cdot f(t) \cdot t = \sum_{y=1} \sum_{t=1}^{g(y)} \tau(t) \cdot t \ . \tag{19}$$

Let $G(t) = \frac{\sum_{d=1}^{t} d \cdot \tau(d)}{\binom{t+1}{2}}$ and note from (18) and (19) that the performance ratio is bounded by $\mathbf{E}[ACS(G)]/opt(G) \leq \max_t G(t)$. As we see in Figure 3, $G$ is concave with a single maxima, with a maximum of 1.8886 attained at $t = 14$. $\square$

4.0.0.5 *Derandomization and weighted graphs.* The algorithm can be derandomized as described in [Halldórsson et al. 2003] and improved in [Epstein et al. 2008]. We describe here an approach based on the latter.

By Theorem 4.7, there exists a value for $\alpha$ that results in the proven expected performance ratio of ACS. Once the sequence $\lfloor q^{i+\alpha} \rfloor_{i=0}$ is fixed, the operation of the algorithm is deterministic.

Each such value of $\alpha$ is one in which some $k_i = \lfloor q^{i+\alpha} \rfloor$ attains a new value, i.e., where $\lfloor q^{i+\alpha} \rfloor = q^{i+\alpha}$. There are only $O(\log_q m)$ different values for $i$, and each involves at most $m$ integers. Hence, it suffices to examine $O(m \log m)$ distinct values for $\alpha$ given as $\log_q x - z$, where $0 \leq x \leq m$ and $0 \leq z \leq \log_q m$.

The analysis above is given for unweighted graphs. It is most easily extended to weighted graphs by viewing each weighted edge as being represented by multiple units all of which are scheduled in the same color. The analysis holds when we view each value $d_r$ (or $\pi_r$) as corresponding to one of these small units. By making these units arbitrarily small, we obtain a matching performance ratio.

Finally, we remark that the algorithm obtains better approximation on non-sparse graphs (albeit, again it is restricted to graph with no parallel edges). Namely, when the average degree is high, the cost-per-edge in the optimal solution is linear in the average degree, while the additional charges will be only logarithmic. Hence, in the case of non-sparse simple graphs, we obtain an improvement over the LP-based approach.

THEOREM 4.9. *The performance ratio of ACS on a simple graph with average degree $\overline{d}$ is at most* $1.79586 + O(\log \overline{d}/\overline{d})$.

PROOF. We first give a lower bound on *opt*. A result of [Bar-Noy et al. 1998] gives that $opt(G) \geq |V(L(G))| + |E(L(G))|/2$ , where $L(G)$ is the line graph of $G$. Note that $|V(L(G))| = m$ and $|E(L(G))| = \sum_{v \in V} \binom{d_v+1}{2} \geq \binom{\overline{d}+1}{2} n = (\overline{d}+1)m$, using concavity. Simplifying, $opt(G) \geq (\overline{d}+3)m/2$.

Recall from (17) that $ACS(G) \leq \sum_{r=1}^{m} \mathbf{E}[\pi_r] \leq \sum_{r=1}^{m} 1.79568 d_r + \log_q d_r$. Since $1.79568x + \log_q x$ is a concave function, the bound on $ACS$ is maximized when all $d_r$ are equal, giving $ACS(G) \leq 1.79568D + m(\log_q(D/m)) \leq 1.79568 opt(G) + m(\log_q(opt(G)/m))$. Hence, using that $\log(x)/x$ is strictly decreasing for $x \geq e$, we have that the performance ratio of $ACS$ is at most

$$\frac{ACS(G)}{opt(G)} \leq 1.79568 + \frac{\log_q(opt(G)/m)}{opt(G)/m} = 1.79568 + \frac{\log_q(\overline{d}+3)/2}{(\overline{d}+3)/2} \ .$$

□

REFERENCES

AFRATI, F. N., BAMPIS, E., FISHKIN, A. V., JANSEN, K., AND KENYON, C. 2000. Scheduling to minimize the average completion time of dedicated tasks. In *Foundations of Software Technology and Theoretical Computer Science, 20th Conference, FST TCS 2000 New Delhi, India, December 13-15, 2000, Proceedings*, S. Kapoor and S. Prasad, Eds. Lecture Notes in Computer Science, vol. 1974. Springer, 454–464.

BANSAL, N. AND SVIRIDENKO, M. 2006. The Santa Claus problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006.* 31–40.

BAR-NOY, A., BELLARE, M., HALLDÓRSSON, M. M., SHACHNAI, H., AND TAMIR, T. 1998. On chromatic sums and distributed resource allocation. *Inf. Comput. 140,* 2, 183–202.

BAR-NOY, A. AND KORTSARZ, G. 1998. Minimum color sum of bipartite graphs. *J. Algorithms 28,* 2, 339–365.

CHARIKAR, M., CHEKURI, C., GOEL, A., GUHA, S., AND PLOTKIN, S. A. 1998. Approximating a finite metric by a small number of tree metrics. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS).* 379–388.

CHEKURI, C. AND KHANNA, S. 2004. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis.* Chapman & Hall/CRC, Chapter Chapter 11: Approximation Algorithms for Minimizing Average Weighted Completion Time.

CHUDAK, F. A. AND SHMOYS, D. B. 2003. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM J. Comput. 33,* 1, 1–25.

CHUZHOY, J., OSTROVSKY, R., AND RABANI, Y. 2006. Approximation algorithms for the job interval selection problem and related scheduling problems. *Math. Oper. Res. 31,* 4, 730–738.

Coffman, Jr, E. G., Garey, M. R., Johnson, D. S., and LaPaugh, A. S. 1985. Scheduling file transfers. *SIAM J. Comput. 14,* 3, 744–780.

Cook, W., Cunningham, W., Pulleyblank, W., and Schrijver., A. 1998. *Combinatorial Optimization*. John Wiley and Sons.

Dobson, G. and Karmarkar, U. S. 1989. Simultaneous resource scheduling to minimize weighted flow times. *Operations Research 37,* 4, pp. 592–600.

Epstein, L., Halldórsson, M. M., Levin, A., and Shachnai, H. 2008. Weighted sum coloring in batch scheduling of conflicting jobs. *Algorithmica 55,* 4, 643–665.

Fishkin, A. V., Jansen, K., and Porkolab, L. 2001. On minimizing average weighted completion time of multiprocessor tasks with release dates. In *Proceedings of the 28th International Colloquium Automata, Languages and Programming, ICALP 2001, Crete, Greece, July 8-12, 2001*, F. Orejas, P. G. Spirakis, and J. van Leeuwen, Eds. Lecture Notes in Computer Science, vol. 2076. Springer, 875–886.

Gandhi, R., Halldórsson, M. M., Kortsarz, G., and Shachnai, H. 2008. Approximating non-preemptive open-shop scheduling and related problems. *ACM Trans. Algorithms 4,* 1.

Gandhi, R. and Mestre, J. 2009. Combinatorial algorithms for data migration to minimize average completion time. *Algorithmica 54,* 1, 54–71.

Gehringer, E. F., Siewiorek, D., and Segall, Z. 1986. *Parallel processing: the Cm* experience*. Digital Press, Newton, MA, USA.

Giaro, K., Kubale, M., Malafiejski, M., and Piwakowski, K. 2002. Dedicated scheduling of biprocessor tasks to minimize mean flow time. In *Proceedings of Parallel Processing and Applied Mathematics, 4th International Conference, PPAM 2001 Naleczow, Poland, September 9-12, 2001, Revised Papers*, R. Wyrzykowski, J. Dongarra, M. Paprzycki, and J. Wasniewski, Eds. Lecture Notes in Computer Science, vol. 2328. Springer, 87–96.

Hall, L. A., Schulz, A. S., Shmoys, D. B., and Wein, J. 1997. Scheduling to minimize average completion time: off-line and on-line approximation algorithms. *Math. Oper. Res. 22*, 513–544.

Halldórsson, M. M. and Kortsarz, G. 2002. Tools for multicoloring with applications to planar graphs and partial $k$-trees. *J. Algorithms 42,* 2, 334–366.

Halldórsson, M. M., Kortsarz, G., and Shachnai, H. 2003. Sum coloring interval and $k$-claw free graphs with application to scheduling dependent jobs. *Algorithmica 37*, 187–209.

Hardy, G. H., Littlewood, J., and Pólya, G. 1952. *Inequalities*, Second edition ed. Cambridge Mathematical Library. Cambridge University Press.

Jain, K. 2001. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica 21,* 1, 39–60.

Khachiyan, L. 1980. Polynomial-time algorithm for linear programming. *USSR Comput. Math. and Math. Physics 20*, 53–72.

Kim, Y. A. 2005. Data migration to minimize the total completion time. *J. Algorithms 55,* 1, 42–57.

Krawczyk, H. and Kubale, M. 1985. An approximation algorithm for diagnostic test scheduling in multicomputer systems. *IEEE Trans. Computers 34,* 9, 869–872.

Kubale, M. 1996. Preemptive versus nonpreemptive scheduling of biprocessor tasks on dedicated processors. *European Journal of Operational Research 94,* 2, 242 – 251.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B. 1993. *Logistics of Production and Inventory*. Handbooks in Operations Research and Management Science, vol. 4. North-Holland, Chapter Sequencing and scheduling: Algorithms and complexity, 445–522.

Malafiejski, M., Giaro, K., Janczewski, R., and Kubale, M. 2004. Sum coloring of bipartite graphs with bounded degree. *Algorithmica 40,* 4, 235–244.

Marx, D. 2004. Minimum sum multicoloring on the edges of planar graphs and partial k-trees. In *Proceedings of the Second International Workshop on Approximation and Online Algorithms, WAOA 2004, Bergen, Norway, September 14-16, 2004*, G. Persiano and R. Solis-Oba, Eds. Lecture Notes in Computer Science, vol. 3351. Springer, 9–22.

Marx, D. 2005. A short proof of the np-completeness of minimum sum interval coloring. *Oper. Res. Lett. 33,* 4, 382–384.

MARX, D. 2006. Minimum sum multicoloring on the edges of trees. *Theor. Comput. Sci. 361,* 2-3, 133–149.

MARX, D. 2009. Complexity results for minimum sum edge coloring. *Discrete Applied Mathematics 157,* 5, 1034–1045.

MESTRE, J. 2008. Adaptive local ratio. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, S.-H. Teng, Ed. SIAM, 152–160.

NUTOV, Z., BENIAMINY, I., AND YUSTER, R. 2006. A (1-1/)-approximation algorithm for the generalized assignment problem. *Oper. Res. Lett. 34,* 3, 283–288.

QUEYRANNE, M. AND SVIRIDENKO, M. 2002a. A (2+epsilon)-approximation algorithm for the generalized preemptive open shop problem with minsum objective. *J. Algorithms 45,* 2, 202–212.

QUEYRANNE, M. AND SVIRIDENKO, M. 2002b. Approximation algorithms for shop scheduling problems with minsum objective. *J. Scheduling 5,* 4, 287–305.

SCHULZ, A. S. AND SKUTELLA, M. 2002. Scheduling unrelated machines by randomized rounding. *SIAM J. Discrete Math. 15,* 4, 450–469.

SVIRIDENKO, M. 2002. An improved approximation algorithm for the metric uncapacitated facility location problem. In *Proceedings of the 9th International Conference on Integer Programming and Combinatorial Optimization, Cambridge, MA, USA, May 27-29.* 240–257.

VIZING, V. G. 1964. On an estimate of the chromatic class of a $p$-graph. *Diskret. Analiz No. 3*, 25–30.

## APPENDIX — Missing Proofs

**Proposition 3.3** *Let $y_1, y_2, \ldots, y_k$ be non-negative numbers satisfying $\sum_{i=1}^{k} y_i = 1$. Then,*

$$\frac{y_1^2}{2} + y_2 \cdot \left(y_1 + \frac{y_2}{2}\right) + y_3 \left(y_1 + y_2 + \frac{y_3}{2}\right) + \ldots + y_k \cdot \sum_{i=1}^{k-1} y_i + \frac{y_k^2}{2} = \frac{1}{2}\ . \tag{20}$$

PROOF. Denote the left hand side by $S$ and rearrange its terms to obtain

$$S = y_1 \left(\frac{y_1}{2} + y_2 + \cdots + y_k\right) + y_2 \left(\frac{y_2}{2} + y_3 + \cdots + y_k\right) + \cdots + y_k \left(\frac{y_k}{2}\right).$$

Applying the equality $\sum_{i=1}^{k} y_i = 1$ to each of the parenthesized sums, we have that

$$\begin{aligned} S &= y_1 \left(1 - \frac{y_1}{2}\right) + y_2 \left(1 - y_1 - \frac{y_2}{2}\right) + \cdots + y_k \left(1 - y_1 - y_2 - \cdots - y_{k-1} - \frac{y_k}{2}\right) \\ &= \sum_{i=1}^{k} y_i - S = 1 - S\ . \end{aligned}$$

Hence, $S$ is $1/2$, as claimed. □

**Proposition 3.4** *Let $t_1, \ldots, t_k$ be positive numbers so that $t_1 \le t_2 \le \cdots \le t_k$, and $x_1, \ldots, x_k$ be positive numbers such that $\sum_{i=1}^{k} x_i \le 1$. Then:*

$$t_1 \cdot x_1 + t_2 \cdot (1 - x_1) \cdot x_2 + \cdots + \prod_{i=1}^{k-1} (1 - x_i) \cdot x_k t_k \le \frac{\left(1 - \prod_{i=1}^{k} (1 - x_i)\right) \cdot \sum_{i=1}^{k} t_i x_i}{\sum_{i=1}^{k} x_i}\ .$$

PROOF. Let $X_i = \sum_{j=1}^{i} x_j$, for $i = 0, \ldots, k$, and let $a = 0$ and $b = X_k$. Define the step functions $f, g : [0, X_k] \to \mathbb{R}$, where $f(x) = t_i$ and $g(x) = \prod_{j=1}^{i-1} (1 - x_j)$

for $x \in [X_{i-1}, X_i)$, $i = 1, 2, \ldots, k$. Then, $\int_a^b f(x)dx = \sum_{i=1}^k x_i t_i$, $\int_a^b g(x)dx = 1 - \prod_{i=1}^k (1 - x_i)$, and $\int_a^b f(x)g(x)dx$ equals the l.h.s. of the claimed inequality. The claim now follows from Chebyshev's sum inequality (see [Hardy et al. 1952, Ineq. 236]), which states that

$$\int_a^b f(x)g(x)dx \le \frac{1}{b-a} \int_a^b f(x)dx \cdot \int_a^b g(x)dx \ ,$$

when $f$ is increasing and $g$ is decreasing. □

**Proposition 3.9** $\left(1 - \left(\prod_{i=1}^{k_h} (1 - \alpha \cdot y_i^h)\right)\right) \le \left(1 + \frac{1}{n}\right) \cdot \left(1 - \frac{1}{e^\alpha}\right)$ .

Proof. The term is maximized when $\prod_{i=1}^{k_h} \left(1 - \alpha \cdot y_i^h\right)$ is minimized. Recall the first inequality of Proposition 3.5:

$$\frac{1}{e^\alpha}\left(1 - \frac{3}{n^2}\right) \le \prod_{i=1}^{k_h} \left(1 - \alpha y_i^h\right) \ .$$

We then get

$$\left(1 - \left(\prod_{i=1}^{k_h} (1 - \alpha \cdot y_i^h)\right)\right) \le \left(1 - \frac{1}{e^\alpha} \cdot \left(1 - \frac{3}{n^2}\right)\right) \le \left(1 + \frac{1}{n}\right) \cdot \left(1 - \frac{1}{e^\alpha}\right) \ .$$

The above is easily reduced to $3/(e^\alpha - 1) \le n$ which holds for large enough $n$ as $\alpha \ge 1/2$. □

**Lemma 4.3** *For each edge $r$,*

$$\pi_r \le \varphi_r \left[\frac{1}{q-1} + \frac{1}{2}\right] + b_r + 1 - \frac{q^\alpha}{q-1} + \gamma_r \ .$$

*where $\varphi_r = q^{b_r + \alpha}$ and*

$$\gamma_r = \frac{\beta_r + 1}{2} \left(\lfloor q^\alpha \rfloor - q^\alpha - \delta\right) \ .$$

Proof. For $r$ for which $b_r > 0$, and thus $\beta_r = 1$, we have

$$\begin{aligned}
\pi_r &= \sum_{i=0}^{b_r - 1} \lfloor q^{i+\alpha} + 1 \rfloor - \delta + \frac{\lfloor q^{b_r+\alpha} \rfloor + 2}{2} \\
&\le (\lfloor q^\alpha \rfloor - q^\alpha) + \sum_{i=0}^{b_r - 1} (q^{i+\alpha} + 1) - \delta + \frac{q^{b_r+\alpha} + 2}{2} \\
&= q^{b_r+\alpha} \left[\frac{1}{q-1} + \frac{1}{2}\right] - \frac{q^\alpha}{q-1} + (b_r + 1) + (\lfloor q^\alpha \rfloor - q^\alpha) - \delta \\
&= \varphi_r \left[\frac{1}{q-1} + \frac{1}{2}\right] + b_r + 1 - \frac{q^\alpha}{q-1} + \gamma_r \ .
\end{aligned}$$

For $r$ with $b_r = 0$, and thus $\beta_r = 0$, we have

$$\begin{aligned}\pi_r &= \frac{\lfloor q^\alpha \rfloor + 2 - \delta}{2} \\ &= q^\alpha \left[\frac{1}{q-1} + \frac{1}{2}\right] - \frac{q^\alpha}{q-1} + 1 + \frac{\lfloor q^\alpha \rfloor - q^\alpha - \delta}{2} \\ &= \varphi_r \left[\frac{1}{q-1} + \frac{1}{2}\right] + b_r + 1 - \frac{q^\alpha}{q-1} + \gamma_r \ .\end{aligned}$$

□

**Lemma 4.6** *For $q \in [3, 4)$,*

$$\mathbf{E}[\gamma_r] = \begin{cases} \frac{1}{2}\left(3 - \log_q 12 - \frac{q-1}{\ln q}\right) & \text{if } d_r = 1 \\ \frac{1}{2}\left(3 - \log_q 12 - \frac{q}{\ln q}\right) & \text{if } d_r = 2 \\ \frac{1}{2}\left(3 - \log_q 16/3 - \frac{q+1}{\ln q}\right) & \text{if } d_r = 3, \text{ and} \\ 3 - \log_q 12 - \frac{q-1}{\ln q} & \text{if } d_r \geq 4. \end{cases}$$

Proof. Let $s$ be a real number such that $d_r = q^{s+\alpha}$. Recall from the proof of Lemma 4.5 that $b_r = \lceil s \rceil = \lceil \log_q d_r - \alpha \rceil$. This can be rewritten as $b_r = \lfloor \log_q d_r \rfloor + t_r$, where $t_r$ is a Bernoulli variable with probability $\log_q d_r - \lfloor \log_q d_r \rfloor$. Thus,

$$\mathbf{E}[b_r] = \log_q d_r \ . \tag{21}$$

Note that

$$\mathbf{E}\left[\lfloor q^\alpha \rfloor\right] = \sum_{i=1}^{\lfloor q \rfloor} i \cdot \int_{\alpha=\log_q i}^{\min(1,\log_q i+1)} d\alpha = \lfloor q \rfloor - \sum_{i=2}^{\lfloor q \rfloor} \log_q i = 3 - \log_q 6, \tag{22}$$

whenever $3 \leq q < 4$.

For $r$ with $d_r \geq 4$, we have that $b_r > 0$, and thus $\beta = 1$. Hence, using (22), we have that

$$\mathbf{E}[\gamma_r] = \mathbf{E}\left[\lfloor q^\alpha \rfloor\right] - \mathbf{E}\left[q^\alpha\right] - \mathbf{E}[\delta] = (3 - \log_q 6) - \frac{q-1}{\ln q} - \log_q 2 = 3 - \log_q 12 - \frac{q-1}{\ln q} \ .$$

For $r$ with $d_r = 1$ it holds that $b_r = \beta_r = 0$. Hence,

$$\mathbf{E}[\gamma_r] = \frac{1}{2}\left(\mathbf{E}\left[\lfloor q^\alpha \rfloor\right] - \mathbf{E}\left[q^\alpha\right] - \mathbf{E}[\delta]\right) = \frac{1}{2}\left(3 - \log_q 12 - \frac{q-1}{\ln q}\right) \ .$$

For the remaining two cases, we need to consider three regions for the value of $\alpha$, with breakpoints at $\log_q 2$ and $\log_q 3$. The variables $\lfloor q^\alpha \rfloor$, $\delta$ and $\beta_r$ are constant in each of these regions. Namely,

$$\lfloor q^\alpha \rfloor = \begin{cases} 1 & \alpha \in [0, \log_q 2) \\ 2 & \alpha \in [\log_q 2, \log_q 3) \\ 3 & \alpha \in [\log_q 3, 1), \end{cases} \quad \delta = \begin{cases} 1 & \alpha \in [0, \log_q 2) \\ 0 & \alpha \in [\log_q 2, 1), \end{cases} \quad \beta_r = \begin{cases} 1 & \alpha \in [0, \min(\log_q d_r, 1)) \\ 0 & \text{otherwise.)} \end{cases}$$

When $d_r = 2$,

$$\begin{aligned}\mathbf{E}[\gamma_r] &= \int_{x=0}^{\log_q 2} (1 - q^x - 1)dx + \int_{x=\log_q 2}^{\log_q 3} \frac{1}{2}(2 - q^x)dx + \int_{x=\log_q 3}^{1} \frac{1}{2}(3 - q^x)dx \\ &= -\left[\frac{q^x}{\ln q}\right]_{x=0}^{\log_q 2} + [x]_{x=\log_q 2}^{\log_q 3} + \left[\frac{3x}{2}\right]_{x=\log_q 3}^{1} - \left[\frac{q^x}{2\ln q}\right]_{x=\log_q 2}^{1} \\ &= \frac{1}{2}\left(3 - \log_q 12 - \frac{q}{\ln q}\right) .\end{aligned}$$

while when $d_r = 3$,

$$\begin{aligned}\mathbf{E}[\gamma_r] &= \int_{x=0}^{\log_q 2} (1 - q^x - 1)dx + \int_{x=\log_q 2}^{\log_q 3} (2 - q^x)dx + \int_{x=\log_q 3}^{1} \frac{1}{2}(3 - q^x)dx \\ &= -\left[\frac{q^x}{\ln q}\right]_{x=0}^{\log_q 3} + [2x]_{\log_q 2}^{\log_q 3} + \frac{1}{2}\left[3x - \frac{q^x}{\ln q}\right]_{\log_q 3}^{1} \\ &= \frac{1}{2}\left(3 - \log_q 16/3 - \frac{q+1}{\ln q}\right) .\end{aligned}$$

□

ACKNOWLEDGMENTS

We wish to thank the anonymous referees for their insightful comments.

Received March 2008; revised February 2009 and August 2010; accepted September 2010.