Approximating Circular Arc Colouring and Bandwidth Allocation in All-Optical Ring Networks

Vijay Kumar

Department of ECE
Northwestern University
Evanston, IL 60208, U.S.A.
vijay@ece.nwu.edu
http://www.ece.nwu.edu/~vijay/

Abstract. We present randomized approximation algorithms for the *circular arc graph colouring problem* and for the problem of *bandwidth allocation* in all-optical ring networks. We obtain a factor-of-$(1 + 1/e + o(1))$ randomized approximation algorithm for the arc colouring problem, an improvement over the best previously known performance ratio of 5/3. For the problem of allocating bandwidth in an all-optical *WDM* (*wavelength division multiplexing*) ring network, we present a factor-of-$(1.5+1/2e+o(1))$ randomized approximation algorithm, improving upon the best previously known performance ratio of 2.

1 Introduction

The *circular arc colouring problem* is the problem of finding a minimal colouring of a set of arcs of a circle such that no two overlapping arcs share a colour. Applications include problems in network design and scheduling. There have been several investigations of the circular arc colouring problem ([17],[5]). The problem was shown to be NP-complete by Garey, Johnson, Miller and Papadimitriou in [5]. Tucker [17] reduced the problem to an integral multicommodity flow problem. For the special case of the *proper* circular arc colouring problem (a set of circular arcs is *proper* if no arc is contained in another), as an $O(n^2)$ algorithm is due to Orlin, Bonuccelli and Bovet [12]. For the general case, Tucker [17] gave a simple approximation algorithm with an approximation ratio of 2. An approximation algorithm with a performance ratio of 5/3 is due to Shih and Hsu [15].

We present a randomized approximation algorithm that achieves a performance ratio of $1 + 1/e + o(1)$ for instances where $d = \Omega(\ln n)$, where d is the minimum number of colours needed and n the number of distinct arc endpoints. Optical networks make it possible to transmit data at very high speeds, of the order of several gigabits per second. Electronic switches can not operate at such high speeds, so to enable data transmission at high speeds, it is necessary to keep the signal in optical form. Such networks are termed *all-optical networks.*

Wavelength division multiplexing (WDM) is a technology which allows for multiple signals to be carried over a link by laser beams with different wavelengths. We can think of these signals as light beams of different colours. As a signal is to be carried by the same beam of light throughout its path, a wavelength needs to be assigned to each connectivity request. To prevent interference, wavelengths must be assigned in such a way that no two paths that share a link are assigned the same wavelength. Several different network topologies have been studied, including trees, rings, trees of rings, and meshes ([14, 8, 10, 3, 7]). Rings are a very common topology: nodes in an area are usually interconnected by means of a ring network. Also, sometimes WDM networks evolve from existing fibre networks such as SONET rings. For the problem of bandwidth allocation in rings, Raghavan and Upfal [14] give an approximation algorithm within twice the optimal. We present an algorithm that has an asymptotic performance ratio of $1.5 + 1/2e + o(1)$, except when the bandwidth requirement is very small. Communication in SONET rings requires establishing point-to-point paths and the allocation of bandwidth to paths in a conflict-free manner (see [2]). The algorithmic aspect of the task is identical to that in WDM networks. Our solution, therefore, extends to this problem as well.

2 Arc Colouring and Multicommodity Flows

We are given a family F of arcs. An *overlap set* is the set of all arcs in F that contain some particular point on the circle. We will refer to the size of the largest overlap set as the *width* of F. Let $p_0, p_1, \cdots p_{n-1}$ be the n distinct endpoints of arcs in F, in clockwise order starting from some arbitrary point on the circle. An arc that runs clockwise from p_i to p_{i+1} for some i, or from p_{n-1} to p_0, is an arc of *unit length*. The *chromatic number* of F, denoted by $\gamma(F)$, is the smallest number of colours required to colour F.

Let d be the width of F. We begin by adding extra arcs of unit length to F in order to get a family of arcs of *uniform width*, that is, one in which all overlap sets are of equal cardinality. Consider a point P on the circle, between some p_i and p_{i+1}. Let there be d' arcs containing P. We add $d - d'$ arcs of unit length to F which run from p_i to p_{i+1}. Doing this for every pair of consecutive endpoints, we obtain a new family F' of arcs which is of uniform width d. This transformation helps simplify the description of our algorithm, and it is straightforward to show that $\gamma(F) = \gamma(F')$. Now suppose we were to take each arc A_i of F' that contains the point p_0, and cut it at p_0 to obtain two arcs: arc A_i^1 beginning at p_0, and arc A_i^2 terminating at p_0. The new family of arcs obtained is equivalent to a set of intervals of the real line, and can be represented as such. Let $P_0, P_1, \cdots, P_n$ be the n endpoints of intervals in such a representation, ordered from left to right. Arcs of the type A_i^1 can be represented as intervals S_i^1 beginning at P_0, while arcs of type A_i^2 can be represented as intervals S_i^2 terminating at P_n. Any other arc runs from p_i to p_j, and does not contain p_0. Such an arc A_i is represented as an interval S_i from P_i to P_j. Let I be the resulting set of intervals. I is of uniform width, that is, there are exactly d intervals passing over any point between P_0

and P_n. I can be partitioned into d unit-width sets of intervals, $I_1, I_2, \ldots, I_d$, such that S_i^1 is contained in I_i. Note that a c-colouring of I in which each S_i^1 gets the same colour as the corresponding S_i^2 is equivalent to a c-colouring of F'. We will refer to such a colouring of I as a *circular colouring.*

Consider a multicommodity network N constructed as follows. The vertices of N are labelled $x_{i,j}$, $i = 1, 2, \ldots, d$; $j = 0, 1, 2, \ldots, 2n-1$. An interval $S_i \in I_j$ originating at P_k and terminating at P_l is represented by an edge from $x_{j,2k}$ to $x_{j,2l-1}$. If some edge terminates at $x_{i,2k-1}$ and another begins at $x_{j,2k}$, for some i, j and k, then we add an edge from $x_{i,2k-1}$ to $x_{j,2k}$. All edges have unit capacity, and an edge can only carry an integral quantity of each commodity. If a vertex has no edges incident on it, it is removed from N. The source s_i for commodity i is located at $x_{i,0}$, and the corresponding destination t_i is the vertex $x_{j,2n-1}$ such that $S_i^2 \in I_j$.

Let us refer to the set of all vertices labelled $x_{i,j}$ for some i as *row* j. We will use the term *column* i to refer to the subgraph of N induced by the set $\{x_{i,j} \mid j = 0, 1, 2, \ldots, 2n-1\}$. We will use the term *layer* i to mean the set of edges that run between or cross over rows $2i$ and $2i+1$. In other words, these are the edges which have exactly one end-point in the set $\{x_{j,k} \mid j \leq 2i\}$.

Note that the number of rows in the network is $2n$. We have added the "extra" n rows for the permuting of colours between columns. Recall that when at a point P some intervals terminate and new intervals begin, the new intervals get a permutation of the colours present on the old intervals.

Lemma 1. *A feasible flow of the d commodities in N is equivalent to a circular d-colouring of I.*

The proof is deferred to the full paper. The network N can be used to decide if the given family of arcs is d-colourable. We can also use a similar technique to decide if it is k-colourable, for any $k > d$. Let F'' be a family of arcs of uniform width k obtained by adding arcs of unit length to F'. It can be easily shown that

Lemma 2. *F' is k-colourable if and only if F'' is k-colourable.*

So to determine the smallest k such that F is k-colourable, we keep adding unit-length arcs to F to obtain a successively larger unit-width family F'' of arcs till we come to a point where the width k of F'' is equal to $\gamma(F'')$. In terms of the multicommodity flow network, this is equivalent to adding extra sources and sinks as well as extra edges. The extra edges help us route the original d commodities. $k-d$ columns will have to be added to the network before a feasible flow can be found. The extra commodities are easy to route. Let N' be the flow network corresponding to F''.

Lemma 3. *If commodities $1, 2, \cdots, d$ can be routed in N' from their sources to the respective destinations, then all the commodities can be routed.*

The proof is omitted for lack of space. Lemma 3 implies that we can modify N' so that it contains the same d sources and d sinks as N. The remaining sources and sinks can be removed. Henceforth, we will consider a network N' with d commodities.

2.1 Approximating the Multicommodity Flow Problem

It is relatively straightforward to set up the multicommodity flow problem as a 0-1 integer program. Soving such a program is NP-complete [4], but relaxing the integrality condition converts it into a linear programming problem which can be solved in polynomial time. We set aside the integrality condition and obtain an optimal solution. Next, we seek to use the information contained in the fractional solution to obtain a good integer solution. To do this, we use a technique called *randomized rounding* [13]. Randomized rounding involves finding a solution to the rational relaxation of an integer problem, and using information derived from that solution to obtain a provably good integer solution.

We begin with a network that has d columns, and keep adding columns till a feasible solution to the LP relaxation is obtained. Let f be a flow obtained by solving the linear program. f can be decomposed into d flows $f_1, f_2, \cdots, f_d$, one for each commodity. Each f_i can further be broken up into a set of paths $P_1, P_2, \cdots, P_p$ from the source of commodity i to its destination. To do this, consider the edge of f_i carrying the smallest amount m_j, and find a source-destination path P_j containing that edge. Associate amount m_j with the path, and subtract amount m_j from the flow carried in f_i by each edge along this path. Repeat this process till no flow remains. This process of breaking a flow into a set of paths is called *path stripping* [13]. Note that $\sum_{j=1}^{p} m_j = 1$.

In order to obtain an integer solution, we will select one path out of these p paths, and use it to route commodity i. To select a path, we cast a p-faced die where $m_1, m_2, \cdots, m_p$ are the probabilities associated with the p faces. Performing such a selection for each commodity, we obtain a set S of d paths to route the d commodities. However, these paths may not constitute a feasible solution since some edge capacity constraints may be violated. Note that in the fractional solution, an edge can carry more than one commodity. It is possible that more than one of these commodities may select a path containing this edge, since the d coin tosses to select the paths are performed independently. However, these conflicts can be resolved by adding extra columns to the network. If there are h paths in S that pass over some edge e, $h-1$ will have to be rerouted.

Consider all the k edges in layer i of the network. There are d paths that use some of these edges. Let d_i edges out of these k edges be contained in some path, $d_i \leq d$. That means that $r_i = d - d_i$ paths will have to be rerouted. Let $r = \max\{r_i\}$. Beginning at the first layer and proceeding layer-by-layer, arbitrarily select the paths to be rerouted in case of conflicts. No more than r paths are selected in any layer. The set of paths obtained can be routed easily by adding r columns to the network. The task is analogous to r-colouring a set of line intervals which has width r.

Thus we get a 0-1 integer flow which routes all the commodities simultaneously. The network has $k+r$ columns, corresponding to a $k+r$ colouring of the family F'' of arcs. A bound on the value of $k+r$ would give us a measure of the goodness of our solution.

2.2 Algorithm Performance

Consider again the edges in layer i of the network. Let E_i be the set of such edges. d_i of these edges are selected by the d commodities in the randomized selection step. In order to minimize r, the number of columns to be added, we need to show that d_i is close to d with high probability.

In the selection step, each commodity randomly chooses a path, thereby selecting one of the k edges of E_i. This is akin to a ball being randomly placed in one of k bins. The situation can be modelled by the classical occupancy problem [11], where d balls are to be randomly and independently distributed into k bins. Let Z be the random variable representing the number of non-empty bins at the end. It can be shown that $\mathbf{E}(Z)$ is minimized when the distribution is uniform. In that case, it is a simple exercise to show that $\mathbf{E}(Z)$ is at least $d - d/e$, where e is the base of the natural logarithm.

To get a high confidence bound on r, we use a famous result in probability theory, called *Azuma's Inequality* [1]. Let $X_0, X_1, \cdots$ be a martingale sequence, and $|X_k - X_{k-1}| \le c_k$ for all k. Azuma's Inequality says that

Theorem 1. *For all $t > 0$ and for any $\lambda > 0$, $\mathbf{Pr}[|X_t - X_0| \ge \lambda] \le 2\exp(-\frac{\lambda^2}{2\sum_{k=1}^{t} c_k^2})$.*

Let X_i denote the expected value of Z after i balls have been distributed. $X_0, X_1, \cdots, X_d$ is a martingale sequence, and application of Azuma's Inequality yields

Corollary 1. $\mathbf{Pr}[|Z - \mathbf{E}(Z)| \ge \lambda\sqrt{d}] \le 2\exp(-\frac{\lambda^2}{2})$

Substituting $\sqrt{4\ln n}$ for λ, we find that with probability $1 - \frac{2}{n^2}$, Z does not deviate from its expected value by more than $2\sqrt{d\ln n}$. That is, with high probability, the number of paths that need to be rerouted due to conflicts at layer i is no more than $d/e + 2\sqrt{d\ln n}$. The probability that more than this many paths need to be rerouted at *any* layer is no more than $n \cdot \frac{2}{n^2} = O(\frac{1}{n})$. So with high probability, r, the number of additional columns, is no more than $d/e + 2\sqrt{d\ln n}$.

So we have a randomized algorithm that finds a feasible integral flow using upto $k + d/e + 2\sqrt{d\ln n}$ network columns. A feasible fractional solution requires k columns, which implies that an integer solution would have at least k columns. In terms of the original arc colouring problem, we obtain a colouring that uses $k + d/e + 2\sqrt{d\ln n}$ colours, where k is a lower bound on the number of colours required by an optimal algorithm. This gives us our result:

Theorem 2. *With high probability, our algorithm can colour a family of arcs of width d using no more than $OPT + d/e + 2\sqrt{d\ln n}$ colours, where n is the number of distinct arc endpoints and OPT the number of colours required by an optimal algorithm.*

For the case where $\ln n = o(d)$, our algorithm has an asymptotic performance ratio of $1 + 1/e$. In most applications, this is the interesting range of values. On the other hand, if d is sufficiently small, it is possible to solve the problem optimally in polynomial time: for the case when $d\ln d = O(\ln n)$, a polynomial time algorithm is presented in [5].

3 The Bandwidth Assignment Problem in Optical Rings

We can represent a ring network as a circle, using points and arcs to denote nodes and paths respectively. Let n denote the number of nodes in the ring. We are given a set of communication *requests*, where each request is a (*source, destination*) pair, and *source* and *destination* are points on the circle. The task is to associate an arc with each request that connects the corresponding source and destination, and assign a colour to each arc in such a way that the set of arcs overlapping any particular point P on the circle does not contain more than one occurence of any colour.

We define an *instance* to be a collection of requests and arcs (uncoloured as yet). That is, some of the requests may have been routed. We say that a collection C of arcs is *derivable* from I if C can be obtained from I by routing the requests of I. Let $D(I)$ denote the collection of all such sets of arcs derivable from I. A *solution* for an instance I is some $C \in D(I)$ together with a valid colouring of C. An optimal solution is the one that uses the fewest colours among all solutions.

When we solve an LP relaxation of the problem, an arc may receive several colours in fractional quantities. If arc a receives quantity x of colour i, we will say that a *fraction of weight* x of a receives colour i. An undivided arc has weight 1. While working with fractional solutions, we will freely split arcs into fractions. We will also extend the notion of weight to intervals of the real line.

We will use the term *interval set* to mean a collection of intervals of the real line. The *cross-section* of an interval set S at point P is the sum of weights of all intervals of S that contain P. The *width* of an interval set S is the largest cross-section of S over all points P. In the case of a set of unsplit intervals, the width is the size of the largest clique in the corresponding interval graph.

A *conflicting pair* of arcs is a pair of arcs (a_1, a_2) such that every point on the circle is contained in at least one of (a_1, a_2), and there is some point on the circle overlapped by both a_1 and a_2. A *parallel routing* is a collection of arcs that does not contain any conflicting pairs. In the following, we examine some interesting and helpful properties of parallel routings.

Let C be a parallel routing, and S_e the set of all arcs in C that contain a link e of the ring.

Lemma 4. *S_e does not contain the whole circle.*

Proof. Assume otherwise. Let a be the arc in S_e whose clockwise endpoint is farthest from e, and let b have the farthest anticlockwise endpoint. Clearly a and b together contain the whole circle, and overlap each other over e. This means that they constitute a conflicting pair, which is not possible in a parallel routing.

As a and b together do not contain the whole circle, some link f is not contained in either:

Lemma 5. *For every link e there is another link f such that no arc of C contains both e and f.*

Each of e and f is a *complement* of the other. The removal of e and f would break the ring into two *halves.* A *complementary bisection* $CB(e, f)$ is a pair of halves created by the removal of two links e and f which are complements of each other.

Lemma 6. *If both the endpoints of an arc lie in the same half of some complementary bisection $CB(e, f)$, then in any parallel routing, that arc is contained entirely in that half.*

The following lemma lets us restrict our attention to parallel routings in the search for an optimal solution. The simple proof is omitted and can be found in the full paper.

Lemma 7. *For any instance I there is an optimal solution Z whose arcs form a parallel routing.*

Define a (c, w) *colour partition* of a family of arcs to be a partition into c families of arcs $C_1, C_2, \cdots, C_c$ each of width 1 and an interval set S of width w. The size of such a partition is defined to be $c + w$. An optimal colour partition is one of minimum size. Colour-partitioning is related to colouring. It is easy to show that

Lemma 8. *An optimal colouring of a family of arcs that uses k colours is equivalent to an optimal colour partition of size k.*

The bandwidth allocation problem can now be looked upon as the problem of routing a set of requests to minimise the size of the optimal colour partition of the resultant routing, and obtaining such a colour partition.

3.1 The Allocation Algorithm

We can route a subset of the requests in I, the given instance, in accordance with Lemmas 6 and 7. We select a link e randomly, and let us assume for the moment that we know a link f with the following property: f is the complement of e in some optimal solution Z_{opt} whose arcs constitute a parallel routing P. According to Lemma 7, such a solution must exist. Consider a request r for which there is a source-destination path p which does not include either of e and f. In Z_{opt}, such a request r must be routed over p, in accordance with Lemma 6. We route every such request r using the corresponding p. Let I' be the instance resulting from such routing. Clearly, P is still derivable from I'.

The remaining requests are the ones whose source and destination lie in different halves of $CB(e, f)$. We will refer to such requests as *crossover* requests. We set up an integer program $IP(I')$ to route the crossover requests of I' to obtain $I'' \in D(I)$ such that the size of the colour partition of I'' is the smallest among all the members of $D(I)$. Our next step is to solve $LP(I')$, the LP relaxation of $IP(I')$ to get an optimal fractional solution Z_f.

Finally, we use information from Z_f to obtain an integer solution Z_I provably close to Z_f. We try to replace the fractional quantities in Z with integer values in such a way that the resulting solution Z_I is a feasible solution to $IP(I')$ and further, with high probability the objective function value of Z_I is close to that of Z_f.

We assumed above that we know the edge f. However, this is not so. We can remedy the situation by running the algorithm for all the possible $n-1$ choices of f and taking the best of the solutions obtained, giving us a solution no worse than the one we would get if we knew the identity of f.

3.2 Integer Programming and LP Relaxation

Let $r_1, r_2, \cdots, r_l$ be the l crossover requests in I'. Each r_i can either be routed as arc b_i, which overlaps link e, or as $\bar{b}_i$, the complementary arc. Let $\{a_1, a_2, \cdots, a_m\}$ be the collection of arcs obtained by taking all the arcs of the kind b_i and $\bar{b}_i$ together with the arcs in I'. Let x_i be the indicator variable that is 1 if r_i is routed as b_i and 0 otherwise. Since all arcs b_i must have distinct colours, we can without loss of generality require that if arc b_i is selected for r_i, it must bear colour i. Otherwise, colour i will not be used. In other words, x_i is the quantity of colour i that is used. If these colours are not sufficient for all the arcs, some arcs are allowed to remain uncoloured. Let $\{a_1, a_2, \cdots, a_m\}$ be the collection of all the arcs, including arcs of the kind b_i or $\bar{b}_i$. Let $y_{i,j}$ be the indicator variable that is 1 if arc a_i gets colour j, and 0 otherwise. $y_{i,0} = 1$ if a_i is uncoloured, and 0 otherwise. To avoid colour conflicts, we require that for each link g and colour j, no more than amount x_j of colour j should be present on all the arcs that contain link g.

A feasible solution to this integer program $IP(I')$ is equivalent to a colour partition, since each colour i is present on a family of arcs of width 1, and the uncoloured arcs constitute an interval set since none of them contain link e. The objective function is therefore set up to minimize the size of this colour partition.

Consider a feasible solution F to $IP(I')$. F contains some $I'' \in D(I)$, since each request of I' has been assigned an arc. Each colour i is present on a family of arcs of width 1, and the uncoloured arcs constitute an interval set since none of them contain link e. Therefore, F contains a colour partition of I''. As the objective function is set up to minimise the size of the colour partition, an optimal solution must represent an I'' with the smallest-sized colour partition among all the members of $D(I)$.

The next step in our algorithm is to obtain an optimal solution to $LP(I')$, the LP relaxation of $IP(I')$. But before we relax the integrality condition and solve the resulting linear program, we will introduce an additional constraint: colour i is not a valid colour for $\bar{b}_i$. This is not required in the integer program since if r_i is routed as $\bar{b}_i$, x_i is 0 and colour i is not used at all. However, with the relaxation it is possible for r_i to be split between b_i and $\bar{b}_i$. We wish to ensure that b_i and $\bar{b}_i$ do not end up sharing colour i. Since the additional constraint is redundant in the integer program, it does not alter the solution space of the integer program, and hence does not change the optimal solution. We solve the linear program to get an optimal fractional solution Z_f.

Our rounding technique requires our fractional solution to satisfy the following property:

Property 1. For each i such that $x_i \neq 0$, the corresponding $\bar{b_i}$ is entirely uncoloured.

It is not possible to express this property as a linear constraint. We modify Z_f by recolouring techniques to obtain a sub-optimal fractional solution Z that satisfies it. A description of the recolouring techniques involved is deferred to the full paper. The recolouring process may cause an increase in the objective function value and a decrease in the total amount of colour used.

The last step in our bandwidth allocation algorithm is the computation of a good integer solution Z_I to the integer program $IP(I')$ formulated above. This involves rounding off the fractional quantities in the fractional solution Z to integer values in a randomized fashion.

Z resembles the solution to the multicommodity flow problem in Section 2.1 and can be looked upon as a collection of flows. Let P be a point in the middle of link e. Quantity x_i of colour i flows from P round the circle and back to P. The flow of colour i can be decomposed into paths $P_1, P_2, \cdots, P_p$, where each P_j is a family of circular arcs of width 1. An amount m_j of colour i is associated with P_j, and $\sum_1^p m_j = x_i$. We associate probability m_j with each P_j, and use a coin toss to select one of them. With probability x_i, some P_k is selected. We use the arcs of P_k to carry a unit amount of colour i. With probability $1 - x_i$ no path is selected, in which case r_i will be routed as arc $\bar{b_i}$ (and may be selected to carry some other colour). We repeat this procedure independently for each colour i. If two or more different colours select some arc a_i, we randomly pick one of them for a_i. If no colour picks a_i, it is added to the interval set of uncoloured arcs.

At the end of this procedure, all fractional quantities have been converted into 0 or 1, and the constraints are still satisfied. Let us see how far this integer solution is from the optimal.

3.3 Algorithm Performance

First of all, the objective function value of the optimal fractional solution Z_f is obviously a lower bound on that of an optimal solution to $IP(I')$. Let us compare our final integer solution Z_I with Z_f. Let z_I, z and z^* be the objective funtion values of Z_I, Z and Z_f respectively, and let $\Delta = z_I - z^*$. In the following, we try to bound Δ. Let $\Delta_1 = z_I - z$ and $\Delta_2 = z - z^*$. Then $\Delta = \Delta_1 + \Delta_2$.

As we mentioned earlier, the objective function value may increase during the recolouring process while the amount of colour used may decrease. Let $c = \sum_1^l x_i$ denote the total amount of colour in Z. Let c_f and c' be the corresponding quantities for Z_f and Z_I respectively. Our recolouring technique achieves the following bound on Δ_2, the cost of recolouring.

Lemma 9. $0 \leq z - z^* \leq c_f - c \leq c_f/2 \leq z^*/2$.

The details are deferred to the full paper. Let us now estimate Δ_1. Let S be the uncoloured interval set contained in Z and let w be its width. Let S' be the uncoloured interval set of width w' in Z_I. Let $c = \sum_1^l x_i$ denote the total amount of colour in Z. Let c' be the corresponding quantity in case of Z_I. The following bounds, due to Hoeffding [9], are useful in estimating $c' - c$. Consider a quantity X that is the sum of n independent Poisson trials, Then

Lemma 10.

$$\mathbf{Pr}[X - \mathbf{E}(X) \geq n\delta] \leq e^{-n\delta^2} \qquad \textit{for} \;\; \delta > 0. \tag{1}$$

$$\mathbf{Pr}[X - \mathbf{E}(X) \geq \delta \mathbf{E}(X)] \leq e^{-\frac{\delta^2}{3}\mathbf{E}(X)} \qquad \textit{where} \;\; 0 < \delta < 1. \tag{2}$$

Let there be m colours in use in the fractional solution Z. Each of these is involved in coin-tossing and path selection.

Lemma 11. $\mathbf{Pr}[c' - c \geq \sqrt{m \ln c}] \leq \frac{1}{c}$

Proof. We can regard c' as a sum of m Poisson trials with associated probabilities x_i. The expected value of c' is c. The result follows from (1) when $\sqrt{\frac{\ln c}{m}}$ is substituted for δ.

Next, let us bound $|w' - w|$. We use the following result due to McDiarmid [9].

Lemma 12. *Let $X_1, X_2, \ldots, X_n$ be independent random variables, with X_i taking values in a set A_i for each i. Suppose that the (measurable) function $f : \Pi A_i \to \mathcal{R}$ satisfies*
$|f(\bar{X}) - f(\bar{X}')| \leq c_i$
whenever the vectors $\bar{X}$ and $\bar{X}'$ differ only in the i^{th} coordinate. Let Y be the random variable $f(X_1, X_2, \ldots, X_n)$. Then for any $\phi > 0$,
$\mathbf{Pr}[|Y - \mathbf{E}(Y)| > \sqrt{\phi \sum_i c_i^2/2}] \leq 2e^{-\phi}$.

Lemma 13. $\mathbf{Pr}[|w' - \mathbf{E}[w']| > \sqrt{m \ln c}] \leq \frac{2}{c}$.

Proof. w' is a function of m inpependent choices. c_i is 1 if $x_i = 1$, and 2 if $0 < x_i < 1$. So $\sum_i c_i^2/2$ is no more than m, and Lemma 12 yields
$\mathbf{Pr}[|w' - \mathbf{E}[w']| > \sqrt{\phi\, m}] \leq 2e^{-\phi}$.
Substituting $\ln c$ for ϕ gives us the desired expression.

Lemmas 14 to 20 seek to bound $\mathbf{E}[w']$. Let $w_p, w_{1,p}$ and $w_{2,p}$ denote the cross-sections of S, S_1 and S_2 respectively at a point p on the ring. Let $w'_p, w'_{1,p}$ and $w'_{2,p}$ be the respective quantities for S', S'_1 and S'_2. Using Lemma 12 it is straightforward to establish the following two bounds.

Lemma 14. $\mathbf{Pr}[|w'_{1,p} - w_{1,p}| > \sqrt{m(\ln n + 2\ln c)}] \leq \frac{2}{nc^2}$.

Lemma 15. $\mathbf{Pr}[|w'_{2,p} - w_{2,p}| > c/e + \sqrt{\frac{m}{2}(\ln n + 2\ln c)}] \leq \frac{2}{nc^2}$.

Lemma 16. $\mathbf{Pr}[|w'_p - w_p| > c/e + (1 + 1/\sqrt{2})\sqrt{m(\ln n + 2\ln c)}] \leq \frac{4}{nc^2}$.

Proof. As $|w'_p - w_p| \leq |w'_{1,p} - w_{1,p}| + |w'_{2,p} - w_{2,p}|$, the result follows directly from Lemmas 14 and 15.

Lemma 16 implies that

Lemma 17. $\mathbf{Pr}[\max_p \{w'_p - w_p\} > c/e + (1 + 1/\sqrt{2})\sqrt{m(\ln n + 2\ln c)}] \leq \frac{4}{c^2}$.

We will use the following lemma, the validity of which follows from the definition of expectation.

Lemma 18. *For any random variable X and value x_0, suppose that* $\mathbf{Pr}[X \geq x_0] \leq p$. *Let x_{max} be the largest possible value of X. Then:*
$\mathbf{E}(X) \leq (1-p)x_0 + px_{max}$.

Lemma 19. $\mathbf{E}(\max_p \{w'_p - w_p\}) \leq c/e + 2\sqrt{m(\ln n + \ln c)} + 8/c$.

Proof. Follows from Lemmas 17 and 18 and the fact that $w'_p - w_p$ can not exceed $2c$.

Lemma 20. $\mathbf{E}(w') \leq w + c/e + 2\sqrt{m(\ln n + 2\ln c)} + 8/c$.

Proof. $w' \leq w + \max_p \{w'_p - w_p\}$, so
$\mathbf{E}(w') \leq w + \mathbf{E}(\max_p \{w'_p - w_p\})$.
This in conjunction with Lemma 19 yields the result.

The following bound on w' follows from Lemmas 13 and 20:

Lemma 21. *With probability at least* $1 - \frac{2}{c}$,
$w' \leq w + c/e + 2\sqrt{m(\ln n + 2\ln c)} + 8/c + \sqrt{m \ln c}$.

Lemma 22. *With probability at least* $1 - \frac{3}{c}$, $\Delta_1 \leq c/e + 2\sqrt{m(\ln n + 2\ln c)} + 8/c + \sqrt{m \ln c}$.

Proof. We know that $\Delta_1 = z_I - z = (c' + w') - (c + w) = (c' - c) + (w' - w)$.
Lemma 11 tells us that with probability at least $1 - \frac{1}{c}$, $c' - c < \sqrt{m \ln c}$. Lemma 21 gives us a similar high probability bound on $w' - w$. Together, they directly imply the above bound on Δ_1.

Lemma 23. *With high probability, z_I is no more than* $z^*(\frac{3}{2} + \frac{1}{2e} + o(1)) + O(\sqrt{z^* \ln n})$.

Proof. $\Delta = \Delta_1 + \Delta_2$. Let $c = c_f - x \cdot z^*$, where c_f is the total amount of colour in Z_f. This means that $c \leq c_f - x \cdot c_f$, since c_f is a lower bound on z^*. Lemma 9 implies that $\Delta_2 \leq x \cdot z^*$, and that x can not be more than $\frac{1}{2}$.
Together with Lemma 22, this implies that with high probability,
$\Delta \leq x \cdot z^* + c/e + 2\sqrt{m(\ln n + 2\ln c)} + 8/c + \sqrt{m \ln c}$.
The expression on the right reduces to $z^*(x - \frac{x}{e} + \frac{1}{e} + o(1)) + O(\sqrt{z^* \ln n})$. Since $\Delta = z_I - z^*$, and x can not exceed $\frac{1}{2}$ (Lemma 9), we have our result.

z_I is the size of the colour partition of I' computed by our algorithm. Since a colour partition of size z_I results in a colouring that uses z_I colours, and since z^*, the value of an optimal solution to $LP(I')$, is a lower bound on the value of an optimal solution to $IP(I')$, we have our main result:

Theorem 3. *With high probability, our algorithm requires no more than* $OPT(1.5 + 1/2e + o(1)) + O(\sqrt{OPT}\ln n)$ *wavelengths to accomodate a given set* I *of communication requests, where* OPT *is the smallest number of wavelengths sufficient for* I.

Some of the latest WDM systems involve over a hundred wavelengths. In such a system, $\sqrt{OPT}\ln n$ is likely to be considerably smaller than OPT, giving us a performance ratio close to $1.5 + 1/2e$. In the case of SONET rings the available bandwidth is often much larger, which means that $\ln n$ is typically $o(OPT)$.

References

1. Azuma, K. Weighted Sum of Certain Dependent Random Variables. *Tohoku Mathematical Journal*, 19:357-367, 1967.
2. Cosares, S., Carpenter, T., and Saniee, I. Static Routing and Slotting of Demand in SONET Rings. Presented at the *TIMS/ORSA Joint National Meeting*, Boston, MA, 1994.
3. Erlebach, T. and Jansen, K. Call Scheduling in Trees, Rings and Meshes. In *Proc. 30th Hawaii International Conf. on System Sciences*, 1997.
4. Even, S., Itai, A., and Shamir, A. On the complexity of timetable and multicommodity flow problems, *SIAM Journal of Computing*, 5(1976),691-703.
5. Garey, M.R., Johnson, D.S., Miller, G.L. and Papadimitriou, C.H. The Complexity of Coloring Circular Arcs and Chords. *SIAM J. Alg. Disc. Meth.*, 1(2):216-227, 1980.
6. Golumbic, M. C., *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980.
7. Kaklamanis, C. and Persiano, P. Efficient wavelength routing on directed fiber trees. In *Proc. 4th Annual European Symposium on Algorithms*, 1996.
8. Kumar, V. and Schwabe, E.J. Improved access to optical bandwidth in trees. In *Proc 8th Annual ACM-SIAM Symp. on Discrete Algorithms*, pp. 437-444, 1997.
9. McDiarmid, C. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics*, volume 141 of *LMS Lecture Notes Series*, pages 148–188. 1989.
10. Mihail, M., Kaklamanis, C., and Rao, S. Efficient Access to Optical Bandwidth. In *Proc IEEE Symp on Foundations of Comp Sci*, pp. 548-557, 1995.
11. Motwani, R., and Raghavan, P. *Randomized Algorithms*, Cambridge University Press, 1995.
12. Orlin, J.B., Bonuccelli, M.A., and Bovet, D.P. An $O(n^2)$ Algorithm for Coloring Proper Circular Arc Graphs. *SIAM J. Alg. Disc. Meth.*, 2(2):88-93, 1981.
13. Raghavan, P. Randomized Rounding and Discrete Ham-Sandwiches: Provably Good Algorithms for Routing and Packing Problems. *PhD Thesis, CS Division, UC Berkeley*, 1986.
14. Raghavan, P., and Upfal, E. Efficient Routing in All-Optical Networks. In *Proc 26th ACM Symp on Theory of Computing*, pp. 134-143, 1994.
15. Shih, W.K. and Hsu, W.L. An Approximation Algorithm for Coloring Circular-Arc Graphs. *SIAM Conference on Discrete Mathematics, 1990.*
16. Settembre, M. and Matera, F. All-optical implementations of high capacity TDMA networks. *Fiber and Integrated Optics*, 12:173-186, 1993.
17. Tucker, A. Coloring a Family of Circular Arcs. *SIAM J. Appl. Math.*, 29(3):493-502, 1975.