Approximating Latin Square Extensions

S. Ravi Kumar[*] Dept. of Computer Science Cornell University

Alexander Russell[†] Dept. of Mathematics M.I.T.

Ravi Sundaram[‡] Lab. for Computer Science M.I.T.

December 13, 1995

Abstract

In this paper, we consider the following question: what is the maximum number of entries that can be added to a partially filled latin square? The decision version of this question is known to be **NP**-complete. We present two approximation algorithms for the optimization version of this question. We first prove that the greedy algorithm achieves a factor of 1/3. We then use insights derived from the linear relaxation of an integer program to obtain an algorithm based on matchings that achieves a better performance guarantee of 1/2. These are the first known polynomial-time approximation algorithms for the latin square completion problem that achieve non-trivial worst-case performance guarantees. Our motivation derives from applications to the problems of lightpath assignment and switch configuration in wavelength routed multihop optical networks.

1 Motivation

1.1 Optical Networks

Developments in fiber-optic networking technology using *wavelength division multiplexing* (WDM) have finally reached the point where it is being considered as the most promising candidate for the next generation of wide-area backbone networks. These are highly flexible networks capable of supporting tens of thousands of users and capable of providing capacities on the order of gigabits-per-second per user [4, 11, 17]. WDM optical networks utilize the large bandwidth available in optical fibers by partitioning it into several channels each at a different optical wavelength [1, 4, 14, 15].

The typical optical network consists of routing nodes interconnected by point-to-point fiber-optic links. Each link supports a certain number of wavelengths. The routing nodes are capable of photonic switching, also known as *dynamic wavelength routing* which involves the setting up of *lightpaths* [3, 5, 21]. A lightpath is an optical path between two nodes on a specific wavelength.

*Supported by ONR Young Investigator Award N00014-93-1-0590. This work was done while the author was visiting M.I.T. email: `ravi@cs.cornell.edu`.

†Supported by an NSF Graduate Fellowship and grants NSF 92-12184, AFOSR F49620-92-J-0125, and DARPA N00014-92-1799. email: `acr@theory.lcs.mit.edu`.

‡Supported by grants NSF 92-12184, AFOSR F49620-92-J-0125, and DARPA N00014-92-1799. email: `koods@theory.lcs.mit.edu`.

The optical switch at a node assigns the wavelengths from an incoming port to an outgoing port. This assignment is changeable and can be controlled electronically.

Conflict-free wavelength routing in wide-area optical networks is achieved by utilizing *latin routers* [2]. These are routing devices that employ the concept of a *latin square* (LS). A latin router with n input ports, n output ports, and n wavelengths is associated with a *partial latin square* (PLS), an $n \times n$ matrix that specifies the wavelength connections from the n input ports to the n output ports. The matrix contains elements from the set $\{0\} \bigcup \{1, 2, \ldots, n\}$ (0 is used as a placeholder to denote emptiness) such that each row and each column never contains an element from the set $\{1, 2, \ldots, n\}$ more than once. (see Figure 1 for an example). A non-zero entry L_{ij} of L implies that the wavelength L_{ij} is routed from input port i to output port j. A zero entry denotes an unassigned entry. An LS is a PLS that has no zero entries.

0	0	4	3
2	4	0	1
3	1	0	4
4	3	1	2

Figure 1: A 4×4 PLS

Reducing the number of unassigned or zero entries in the PLS associated with a router is of the utmost practical importance in optical networks as this ensures reduced wastage of the valuable resources of ports and wavelengths. This motivates the following definitions:

Definition 1 *A PLS S_1 is said to* extend *or be an* extension *of a PLS S_2 if S_1 can be obtained by altering only zero entries of S_2.*

Definition 2 *A PLS is said to be* completable *if it can be extended to an LS.*

See Figure 2 for an LS obtained by extending the PLS of Figure 1. Not all PLSs can be completed (see Figure 3).

1	2	4	3
2	4	3	1
3	1	2	4
4	3	1	2

Figure 2: A 4×4 LS

Definition 3 Partial Latin Square Extension Problem (PLSE)*: Given a PLS S_1 find the largest number of zero entries that can be changed to obtain a PLS S_2 that is an extension of S_1.*

The PLSE problem as stated above is an optimization problem. The natural decision version of the problem – namely, given a PLS establish whether it is completable – has been shown to be **NP**-complete [6]. We present the first known polynomial-time approximation algorithms for the PLSE problem with nontrivial worst-case performance guarantees.

1.2 Other Applications

This study also has applications to the more classical areas of statistical designs and error-correcting codes (which were in fact the original drivers of the research into LSs). We refer the interested reader to the (extensive) literature that exists on the subject [7, 8].

2 Previous Work

The subject of LSs has been extensively developed by many eminent combinatorialists. Some of the most famous conjectures concerning LSs were in fact proposed by no less than Euler himself. Denes and Keedwell, [7, 8], provide comprehensive and encyclopedic collections of results on the combinatorial aspects of LSs. Of special interest to us are results concerning the completion of PLSs. The most famous conjecture in this area was the Evans conjecture [9] which was proved after a period of over 20 years by Smetaniuk [19]. An excellent survey of the ongoing attempt to characterize completable PLSs appears in [16].

The computational aspect of completing PLSs was initiated by Rosa [18] and Giles, Oyama and Trotter [10]. The issue was finally resolved by Colbourn [6] who proved that the problem of deciding whether a PLS is completable is **NP**-complete.

Barry and Humblet [2] were the first to recognize the applicability of LSs to the problem of wavelength assignment in optical networks. The question of approximating the PLSE problem was considered at great length by Chen and Banerjee in [3]. They provide an algorithm and a heuristic for approximating the PLSE problem. The algorithm, however, takes exponential time in the worst case. And the heuristic in certain cases could modify the pre-existing entries in the PLS thus rendering it unfit for use in many situations of practical interest.

The rest of the paper is organized as follows: Section 3 contains notation and some basic lemmas; Section 4 contains the factor 1/3 approximation algorithms; Section 5 contains the factor 1/2 approximation algorithms; Section 6 answers some natural questions regarding extensions of certain PLSs; and Section 7 closes with a conjecture that we would be interested in seeing settled.

3 Preliminaries

3.1 Definitions and Notations

Let L be a PLS. If $L_{i,j} = 0$, we say the cell (i,j) is *empty.* Conversely, if $L_{i,j} \neq 0$, we say the cell (i,j) is *filled.* Two PLSs L and M are said to be *compatible* if

- $\forall i,j, L_{ij} = 0$ or $M_{ij} = 0$, and
- $L + M$ is a PLS.

When L and M are compatible LSs we shall denote $L + M$ by $L \oplus M$. For a PLS L, let $|L|$ denote the number of non-empty cells of L. We write $L \subseteq M$ ($L \subset M$) for two PLSs when $M = L \oplus A$ for some (non-trivial) PLS A. This is equivalent to saying that L may be extended to M. We call L *blocked* if $\nexists L' \supset L$. For PLS L, define $L^{\perp}$ to be a compatible LS such that $|L^{\perp}|$ is the largest over all compatible LSs.

The problem of extending a PLS can also be viewed graph-theoretically as a coloring problem. Associate with an $n \times n$ PLS L the colored graph with n^2 vertices $\langle i,j \rangle, 1 \leq i,j \leq n$ and edges $\{(\langle i,j \rangle, \langle i',j' \rangle) | i = i' \text{ or } j = j'\}$ such that vertex $\langle i,j \rangle$ is assigned color $L_{ij} \neq 0$; vertices corresponding to zero entries of L are considered to be uncolored. The problem of PLS extension can now be viewed equivalently as the problem of coloring additional vertices given the corresponding partially colored graph. This motivates our use of the terminology *color* for the entries of a PLS.

3.2 Extending and Completing PLSs - some Combinatorial Lemmas

Colbourn's result, [6], showing that PLS-completability is **NP**-complete has effectively put paid to our hopes of discovering a polynomial time algorithm for recognizing completable PLSs. But it remains an intriguing problem to understand what can be salvaged. We take a combinatorial step in this direction by providing a quantitative characterization of minimally non-completable PLSs and minimally non-extendable or blocked PLSs.

Definition 4 *Let $f(n)$ be the largest number such that every $n \times n$ PLS L with $|L| \leq f(n)$ is completable.*

Lemma 1 $f(n) = n - 1$.

Proof: The Evans conjecture, ([9]) made in 1960, states that any $n \times n$ PLS L with $|L| \leq n - 1$ is completable. It was finally settled in the affirmative by Smetaniuk ([19]) in 1981. This gives us that $f(n) \geq n - 1$. That $f(n) < n$ is easily seen by the PLSs of Figure 3 which cannot be completed. Hence $f(n) = n - 1$. $\square$

Definition 5 *Let $g(n)$ be the largest number such that every $n \times n$ PLS L with $|L| \leq g(n)$ is extendable.*

Lemma 2 $g(n) = \lceil \frac{n^2}{2} \rceil - 1$.

Proof: We first show that $g(n) \geq \frac{n^2}{2} \geq \lceil \frac{n^2}{2} \rceil - 1$. Consider any $n \times n$ PLS L such that $|L| < \frac{n^2}{2}$. Let r_i (c_j)be the set of non-zero entries in row i (column j) of L. If we show that there exists an i, j such that $L_{ij} = 0$ and $|r_i| + |c_j| < n$, then we are done because it implies that L can extended by setting L_{ij} to a value in $\{1, 2, \ldots, n\} - r_i - c_j$. It remains to show that there exists an i, j such that $L_{ij} = 0$ and $|r_i| + |c_j| < n$. We do this by invoking the Cauchy-Schwartz inequality to show that the expectation

$$
\begin{aligned}
E[n - |r_i| + n - |c_j| : L_{ij} = 0] &= \left(\tfrac{1}{n^2 - |L|}\right)\left(\textstyle\sum_i (n - |r_i|)^2 + \sum_j (n - |c_j|)^2\right) \\
&\geq \left(\tfrac{1}{n^2 - |L|}\right)\left(\tfrac{(n^2 - |L|)^2}{n} + \tfrac{(n^2 - |L|)^2}{n}\right) \\
&= 2\left(n - \tfrac{|L|}{n}\right) \\
&> n
\end{aligned}
$$

It is easy to see that $g(n) < \lceil \frac{n^2}{2} \rceil$ by considering the general versions of the examples in Figure 4.

$\square$

1	2	...	n-1	
				n

1			
	1		
		$\ddots$	
			2

Figure 3: Blocked $n \times n$ LS with n entries

1	2		
2	1		
		3	4
		4	3

1	2			
2	1			
		3	4	5
		5	3	4
		4	5	3

Figure 4: Blocked LS with $\lceil \frac{n^2}{2} \rceil$ entries

4 Greedy Algorithms

4.1 A Greedy Algorithm Based on Linear Programming

The problem of maximally extending a PLS L may be expressed as an integer program:

$$\begin{aligned}
&\max \sum_{ijk} x_{ijk} \text{ subject to} \\
&\quad \forall j,k \sum_i x_{ijk} \leq 1 \\
&\quad \forall i,k \sum_j x_{ijk} \leq 1 \\
&\quad \forall i,j \sum_k x_{ijk} \leq 1 \\
&\quad \forall i,j,k \text{ such that } L_{ij} = k \neq 0, x_{ijk} = 1 \qquad (1) \\
&\quad \forall i,j,k, x_{ijk} \in \{0,1\}.
\end{aligned}$$

The association of a feasible point m with the PLS $M_{ij} = k \iff m_{ijk} = 1$ is a natural correspondence between those LSs which extend L and the feasible points of the integer program. A variable which shares no constraint with any variable of (1) shall be called *free*.

Relaxing this integer program to a linear program yields a natural greedy algorithm:

GREEDY (LP):

1. Set $t := 0$. Set $A^0_{ij} = 0$.

2. If $L \oplus A^t$ is blocked, return A^t. Otherwise let x^* be an optimal solution to the linear program for $L \oplus A^t$ and $(\hat{\imath}, \hat{\jmath}, \hat{k})$ be so that $x^*_{\hat{\imath}\hat{\jmath}\hat{k}}$ is a maximum free variable. Set

$$A^{t+1}_{ij} = \begin{cases} \hat{k} & \text{if } (i,j) = (\hat{\imath}, \hat{\jmath}) \\ A^t_{ij} & \text{otherwise.} \end{cases}$$

Increment t and begin step 2 again.

If L is not blocked there exists a free variable so that step 2 may proceed. Furthermore, if x^*_{ijk} is free, k is a consistent assignment to cell L_{ij} so that each augmentation made by step 2 results in a (larger) PLS $L \oplus A^{t+1}$.

Lemma 3 *Let L be a PLS and t the number of iterations performed by* GREEDY (LP) *on L. Then $t \geq (\frac{1}{3-\frac{2}{n}})|L^\perp|$.*

Proof: For a PLS L, let ϕ_L be the optimal value of the associated linear program. Notice that during each iteration of the algorithm, at least one constraint containing a free variable is tight, so that the $x_{\hat{\imath}\hat{\jmath}\hat{k}}$ selected has value at least $1/n$. Examine iteration s of the algorithm. Let x^* be an optimal solution to the linear program for $L \oplus A^s$ having value $\phi_{L\oplus A^s}$ and $(\hat{\imath}, \hat{\jmath}, \hat{k})$ the triple selected in this iteration. Define

$$a_{ijk} = \begin{cases} 1 & \text{if } (i,j,k) = (\hat{\imath}, \hat{\jmath}, \hat{k}) \\ 0 & \text{if } i \neq \hat{\imath} \text{ and } (j,k) = (\hat{\jmath}, \hat{k}) \\ 0 & \text{if } j \neq \hat{\jmath} \text{ and } (i,k) = (\hat{\imath}, \hat{k}) \\ 0 & \text{if } k \neq \hat{k} \text{ and } (i,j) = (\hat{\imath}, \hat{\jmath}) \\ x^*_{ijk} & \text{otherwise.} \end{cases}$$

Notice that a is a feasible solution to the linear program for $L \oplus A^{s+1}$ with value at least $\phi_{L\oplus A^s} - (2 - \frac{2}{n})$. Each iteration, then, depresses the objective function of the associated linear program by at most $(2 - \frac{2}{n})$ whence

$$t \geq \frac{\phi_L - |L|}{3 + \frac{2}{n}} \geq \frac{|L^\perp|}{3 + \frac{2}{n}}.$$

□

Recall that $|A^t| = t$ so that GREEDY (LP) achieves a $\frac{1}{3-\frac{2}{n}}$ approximation factor. This proves the following theorem.

Theorem 4 GREEDY (LP) *is a $\frac{1}{3} + \Omega(\frac{1}{n})$ approximation algorithm.*

4.2 The Naive Greedy Algorithm

Lemma 5 *Let L be a PLS and A, B two PLSs, each compatible with L, so that $L \oplus B$ is blocked. Then $|B| \geq \frac{1}{3}|A|$.*

Proof: For each pair (i,j) with $B_{ij} \neq 0$, let

$$S_{ij} = \{(i,j)\} \cup \{(i,j') \mid B_{ij} = A_{ij'}\} \cup \{(i',j) \mid B_{ij} = A_{i'j}\}.$$

Then $|S_{ij}| \leq 3$. If $|A| > \sum_{ij} |S_{ij}|$ then there is a pair (u,v), appearing in no S_{ij}, so that A_{uv} is non-empty. In this case, $(L \oplus B)_{uv}$ may be consistently set to A_{uv}, contradicting that $L \oplus B$ is blocked. Hence $|A| \leq \sum_{ij} |S_{ij}| \leq 3|B|$. □

Consider the greedy algorithm defined as follows:

GREEDY:

1. Set $t := 0$. Set $A^0_{ij} = 0$.

2. If $L \oplus A^t$ is blocked, return A^t. Otherwise, select a pair $(\hat{\imath},\hat{\jmath})$ with $(L \oplus A^t)_{\hat{\imath}\hat{\jmath}} = 0$ and a color $\hat{k}$ so that
$$A^{t+1}_{ij} = \begin{cases} \hat{k} & \text{if } (i,j) = (\hat{\imath},\hat{\jmath}) \\ A^t_{ij} & \text{otherwise.} \end{cases}$$
is compatible with L. Increment t and begin step 2 again.

Since GREEDY computes an extension A^k so that $L \oplus A^k$ is blocked, $|A^k| \geq \frac{1}{3}|L^\perp|$. This proves the following theorem.

Theorem 6 GREEDY *is a* $\frac{1}{3}$*-approximation algorithm.*

The following example (Figure 5) demonstrates that our analysis of the performance of the greedy algorithm is tight. This PLS can be filled to completion. However, an incorrect choice by GREEDY to fill 2 in $(1,1)$ blocks the LS.

		3	4
	3	4	1
4	1	2	3
3	4	1	1

Figure 5: Worst-case Scenario for the Naive Greedy Algorithm

The greedy algorithm can be implemented very efficiently in practice. For each row i (column j), maintain an n-length bit vector R_i (C_j) marking the empty cells in that row (column). The greedy algorithm goes through each empty square (i,j), picks a number to be filled (if possible) using R_i, C_j, and updates R_i, C_j to reflect the change. This single operation takes $O(n)$ time, yielding an overall running time of $O(n^3)$.

5 Approximation Algorithms Based on Matching

5.1 A Linear Programming Based Algorithm Using Matching

We again consider the linear program associated with a PLS L.

MATCHING (LP):

1. Set $A^0_{ij} = 0$. Carry out the following for $k \in \{1, \ldots, n\}$. Let x^* be a solution to the linear program associated with $L \oplus A^{k-1}$. If $\forall i, j\, x^*_{ijk} = 0$, define $A^k = A^{k-1}$ and move on to the next k. Construct the weighted bipartite graph $G = (U, V, E, w : E \to \mathbb{Q}^+)$ with $U = V = \{1, \ldots, n\}$, $E = \{(u, v) \mid x^*_{uvk} \neq 0\}$, $w(u, v) = x^*_{uvk}$. Select a matching M which maximizes $\frac{|M|}{|M| + \|G\| - \|M\|}$ where $|M|$ is the cardinality of the matching, $\|M\|$ is the weight of the matching, and $\|G\| = \sum_{e \in E} w(e) = \sum_{ij} x^*_{ijk}$ is the total weight of G. Since M is a matching, the variables associated with the edges of M are independent (that is, none of these variables occur together in a constraint) and we may define
$$A^k_{ij} = \begin{cases} k & \text{if } (i, j) \in M \\ A^{k-1}_{ij} & \text{otherwise.} \end{cases}$$
Furthermore, each edge of M corresponds to a non-zero variable so that L and A^k are compatible.

2. Return A^k.

Notice that a matching optimizing the quantity $\frac{|M|}{|M| + \|G\| - \|M\|}$ may be computed in polynomial time by computing a maximum weight matching of each cardinality $c \in \{1, \ldots, n\}$ for which a matching exists and selecting the optimum (see [20], for example). Hence the algorithm runs in polynomial time.

We use the following lemma to analyze the algorithm:

Lemma 7 *Let $G = (U, V, E, w : E \to \mathbb{Q}^+)$ be a weighted bipartite graph with $\forall u, \sum_v w(u, v) \leq 1$. Then for any maximum cardinality matching M, $|M| \geq \|G\|$. Hence $\max_M \frac{|M|}{|M| + \|G\| - \|M\|} \geq \frac{1}{2}$.*

Proof: To begin with, we show that for any maximum matching M in G, there is a subset of vertices W such that:

(i) W covers each edge in E, and

(ii) each edge in M is covered by exactly one vertex in W.

It is easy to see that picking either of the vertices of every edge in M always satisfies the second requirement trivially. Suppose the first requirement is not met. In other words, an edge $(u_0, v_1) \notin M$ is not covered by any vertex in the current W. By maximality of M, there is some $(u_1, v_1) \in M$ such that $u_1 \in W$, by condition (ii). Now, let $W = W \backslash \{u_1\} \cup \{v_1\}$. If the first condition is met, we are done. Otherwise, it implies there is an edge $(u_1, v_2) \notin M$ such that it is not covered by the current W. We repeat the same process now. It is clear that we cannot go indefinitely. When we terminate, we see that $(u_0, v_1), (v_1, u_1), \ldots, (u_{k-1}, v_k)$ is an augmenting path, contradicting the maximality of M.

Since for any vertex u, $\sum_v w(u, v) \leq 1$, the above shows that $|M| \geq \|G\|$. $\Box$

We now consider the effect that stage t of the above algorithm has had on the optimal solution to the linear program. Let $\phi_{L \oplus A^{t-1}}$ be the optimal value of the linear program associated with

$L \oplus A^{t-1}$ and x^* a vector achieving this optimal value. Consider the vector

$$a_{ijk} = \begin{cases} 1 & \text{if } k = t \text{ and } (i,j) \in M \\ 0 & \text{if } k = t \text{ and } (i,j) \notin M \\ 0 & \text{if } k \neq t \text{ and } (i,j) \in M \\ x^*_{ijk} & \text{otherwise.} \end{cases}$$

a is a feasible solution to the linear program associated with $L \oplus A^t$ and

$$\sum_{ijk} a_{ijk} \geq \phi_{L \oplus A^{t-1}} - \|G\| + \|M\|.$$

Hence $\phi_{L\oplus A^t} \geq \phi_{L\oplus A^{t-1}} - \|G\| + \|M\|$. In this case we have set $|M|$ variables and depressed the optimum value of the linear program by at most $\|G\| - \|M\|$. From the above lemma $\frac{|M|}{|M|+\|G\|-\|M\|} > \frac{1}{2}$, so that the above algorithm is a $\frac{1}{2}$-approximation algorithm.

5.2 A Combinatorial Algorithm Using Matching

Consider a PLS L and define $L^t = \{(i,j) \mid \forall s\, L_{is} \neq k, \forall s\, L_{sj} \neq k\}$ to be the collection of cells which will admit a t. Consider the following algorithm:

MATCHING:

1. Set $A^0_{ij} = 0$.
2. For each $k \in \{1, \ldots, n\}$, consider the bipartite graph $G = (U, V, E)$ with $U = V = \{1, \ldots, n\}$ and $E = L^k$. Let M be maximum matching in G. Set

$$A^k_{ij} = \begin{cases} k & \text{if } (i,j) \in M \\ A^{k-1}_{ij} & \text{otherwise.} \end{cases}$$

3. Return A^n.

Consider stage k of the above algorithm. Define

$$P^k_{ij} = \begin{cases} 0 & \text{if } (L \oplus A^{k-1})^\perp_{ij} = k \\ 0 & \text{if } (i,j) \in M \\ (L \oplus A^{k-1})^\perp_{ij} & \text{otherwise.} \end{cases}$$

Notice that P^k is always compatible with $L \oplus A^k$ so that $|(L\oplus A^k)^\perp| \geq |P^k| \geq |(L\oplus A^{k-1})^\perp| - 2|M|$. (Since M is a maximum matching, $(L \oplus A^{k-1})^\perp$ can have no more than $|M|$ cells assigned to k.) This proves the following lemma.

Lemma 8 MATCHING *is a $\frac{1}{2}$-approximation algorithm.*

The following example (Figure 6) demonstrates that our analysis of the performance of the MATCHING algorithm is in fact tight. The PLS (left) can in fact be filled to completion (right), but a bad choice of matching can block it (middle).

We repeat the matching step for each of the n colors. Each matching step can be performed in $O(n^{2.5})$ by the Hopcroft-Karp algorithm ([13]). Therefore, this algorithm runs in $O(n^{3.5})$ time.

	2	3	
2			1
3			2
	1	2	

4	2	3	
2	4		1
3		4	2
	1	2	4

1	2	3	4
2	3	4	1
3	4	1	2
4	1	2	3

Figure 6: Worst-case Scenario for the Matching Algorithm

6 Extending Blocked PLSs

In many applications, the problem of completing a blocked PLS with new available colors is significant. A natural question is this: given a blocked $n \times n$ PLS L, how many extra colors are necessary to complete it. This can be answered exactly (in polynomial time) by constructing the bipartite graph G_L on the $2n$ vertices, $R_i, C_j, 1 \leq i, j \leq n$, such that there is an edge between R_i and C_j iff $L_{ij} = 0$; and observing by Hall's theorem ([12]) that the edge set of this bipartite graph can be partitioned into k^* disjoint matchings where k^* is the maximum degree of a vertex in the bipartite graph defined above. By coloring each such matching with a new color, we ensure that there are no conflicts generated. Thus, k^* new colors suffice. Notice that k^* colors are indeed necessary since some node of G_L has degree k^*.

In fact, one can see that $k^* \leq n/2$ by a proof similar to that of Lemma 2. And, in fact k^* can be equal to $n/2$, as can be seen from the example in Figure 4.

A related question is this: given a blocked L, and k new colors, what is the maximum number of entries that can be filled using these new colors. For $k = 1$, it is equivalent to finding the maximum matching in G_L (defined above) and hence can be exactly computed. For $k > 1$, this problem is equivalent to finding disjoint matchings $M_1, \ldots, M_k$ in G_L such that $\sum_{i=1}^{k} |M_i|$ is maximized. This number can be exactly computed by computing a maximum flow on the following graph G'_L. $V(G_{L'}) = V(G_L) \cup \{s, t\}$, $E(G_{L'}) = E(G_L) \bigcup \cup_{i=1}^{n} (s, R_i) \bigcup \cup_{j=1}^{n} (C_j, t)$, and $c(e) = 1$ if $e \in E(G_L)$ and $c(e) = k$ otherwise. It is easy to extract the actual color assignment to edges from the maximum flow graph.

7 Further Work

Define the *latin square polytope* to be

$$\mathfrak{L}_n = \{x \in (\mathbb{R}^n)^3 \mid \forall i, j, k \; x_{ijk} \geq 0, \forall j, k \sum_i x_{ijk} \leq 1, \forall i, k \sum_j x_{ijk} \leq 1, \forall i, j \sum_k x_{ijk} \leq 1\}.$$

We conjecture the following:

Conjecture 1 *For every vertex* $v \in \mathfrak{L}_n$, $\forall i, j, k, v_{ijk} = 0$ *or* $v_{ijk} \geq \frac{1}{n}$.

This would show that the approximation algorithm of Section 5.1 achieves a factor of $\frac{1}{2} + \Omega(\frac{1}{n})$.

Acknowledgements

We thank Michel Goemans (M.I.T.), Monika Rauch Henzinger (Cornell), Eric Jordan (M.I.T.), Ram Ramanathan (BBN Systems), and Mike Sipser (M.I.T.) for their technical help and suggestions.

References

[1] R. A. Barry and P. A. Humblet. Bounds on the number of wavelengths needed in WDM networks. In *LEOS '92 Summer Topical Meeting Digest*, pages 21–22, 1992.

[2] R. A. Barry and P. A. Humblet. Latin routers, design and implementation. *IEEE/OSA Journal of Lightwave Technology*, pages 891–899, May/June 1993.

[3] C. Chen and S. Banerjee. Optical switch configuration and lightpath assignment in wavelength routing multihop lightwave networks. In *Proceedings of INFOCOM 1995*, pages 1300–1307. IEEE, 1995.

[4] N. K. Cheung, K. Nosu, and G. Winzer, editors. *IEEE JSAC: Special Issue on Dense WDM Networks*, volume 8, August 1990.

[5] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical WANs. *IEEE Transactions on Communication*, 40(7):1171–1182, July 1992.

[6] C. J. Colbourn. The complexity of completing partial latin squares. *Discrete Applied Mathematics*, 8:25–30, 1984.

[7] J. Denes and A. D. Keedwell. *Latin Squares and Their Applications.* Academic Press, Inc., New York, 1974.

[8] J. Denes and A. D. Keedwell. *Latin Squares: New Developments in the Theory and Applications.* Number 46 in Annals of Discrete Mathematics. North-Holland, 1991.

[9] T. Evans. Embedding incomplete latin squares. *Amer. Math. Monthly*, 67:958–961, 1960.

[10] F. R. Giles, T. Oyama, and L. E. Trotter. On completing partial latin squares. In *Proceedings of the Eighth Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 523–543, 1977.

[11] P. E. Green. *Fiber-Optic Networks.* Prentice-Hall, 1992.

[12] M. Hall. Distinct representatives of subsets. *Bulletin of the American Mathematical Society*, 54:922–926, 1948.

[13] J. E. Hopcroft and R. M. Karp. An $n^{\frac{5}{2}}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal of Computing*, 2(4):225–231, December 1973.

[14] IEEE/OSA. *IEEE/OSA Journal of Lightwave Technology, special issue on Broad-Band Optical Networks*, volume 11, May/June 1993.

[15] M. Irshid and M. Kavehrad. A wdm cross-connected star topology for multihop lightwave networks. *IEEE/OSA Journal of Lightwave Technology*, pages 828–835, June 1992.

[16] C. C. Lindner and T. Evans. *Finite Embedding Theorems for Partial Designs and Algebras.* Les Presses de L'Universite de Montreal, 1977.

[17] R. Ramaswami. Multi-wavelength lightwave networks for computer communication. *IEEE Communications Magazine*, 31(2):78–88, February 1993.

[18] A. Rosa. Problem session. In *Proceedings of the Second Caribbean Conference on Combinatorics and Computing*, 1977.

[19] B. Smetaniuk. A new construction of latin squares I. A proof of the Evans conjecture. *Ars Combinatorica*, 11:155–172, 1981.

[20] R. E. Tarjan. *Data Structures and Network Algorithms*, volume 44 of *Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, 1983.

[21] Z. Zhang and A. Acampora. A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength reuse. In *Proc. IEEE Infocom '94*, pages 534–543, Toronto, June 1994.