On Completing Latin Squares

Iman Hajirasouliha*, Hossein Jowhari*, Ravi Kumar**, and Ravi Sundaram***

Abstract. We present a $(\frac{2}{3} - o(1))$-approximation algorithm for the partial latin square extension (PLSE) problem. This improves the current best bound of $1 - \frac{1}{e}$ due to Gomes, Regis, and Shmoys [5]. We also show that PLSE is APX-hard.

We then consider two new and natural variants of PLSE. In the first, there is an added restriction that at most k colors are to be used in the extension; for this problem, we prove a tight approximation threshold of $1 - \frac{1}{e}$. In the second, the goal is to find the largest partial latin square embedded in the given partial latin square that can be extended to completion; we obtain a $\frac{1}{4}$ approximation algorithm in this case.

1 Introduction

Latin squares are elementary combinatorial objects that have been studied for a long time. Informally, a latin square is an $n \times n$ grid, where each cell is filled with a number in $\{1, \ldots, n\}$ and each number occurs exactly once in every row and every column. A partially filled latin square (PLS) is an $n \times n$ grid, where each cell is either empty or filled with a number in $\{1, \ldots, n\}$ and each number occurs at most once in every row and every column. Besides being interesting objects from a mathematical point of view, PLSs have found applications in statistical design, error-correcting codes, and more recently, optical routing. Sudoku puzzles, one of the current fads, are PLSs with additional properties.

To motivate an algorithmic study of PLSs, consider their applications in optical routers [1]. Routers in an optical network are connected by fiber optic links that support a certain number of wavelengths. Each router has some input and output links and is capable of switching wavelengths to avoid conflicts in fiber links. Suppose the router has n input and n output ports and each link can carry n different wavelengths. The snapshot of an active router can be modeled by a PLS as follows. Associate each input port with a row and each output port with a column in the PLS and consider a light signal that comes from the input port i and is routed to the output port j with the new wavelength of k. This can be reflected by assigning k to the cell (i, j) in the PLS.

The question of how much can we increase the utilization of the router is precisely the problem of assigning numbers to the empty cells in a PLS; this is the PLS extension problem (PLSE). Colbourn [2] showed that the decision version

* Simon Fraser University, {ihajiras,hjowahari}@cs.sfu.ca
** Yahoo! Research, ravikumar@yahoo-inc.com
*** Northeastern University, koods@ccs.neu.edu

of PLSE is NP-complete. Kumar, Russell, and Sundaram [9] presented two approximation algorithms for PLSE that achieves factors $\frac{1}{3}$ and $\frac{1}{2}$. Gomes, Regis, and Shmoys [5] obtained an LP-based approximation algorithm that achieves a factor $1 - \frac{1}{e}$, which is currently the best known.

We also consider two natural variants of the PLSE problem. In the k-PLSE problem, the goal is to use at most k different numbers to fill the empty cells in a PLS. This problem arises in optical routers when we wish to invest in at most k new wavelengths, say because of resource considerations. In the c-PLSE problem, the goal is to find the largest PLS embedded in the given PLS that can be extended to completion. This problem arises naturally when we wish to build out an existing network to completion while retaining as much of the existing infrastructure as possible. To the best of our knowledge, neither k-PLSE nor c-PLSE has been studied before.

1.1 Main results

We obtain a $\frac{2}{3} - o(1)$-approximation algorithm for the PLSE problem. This improves the current best bound of $1 - \frac{1}{e}$ due to Gomes, Regis, and Shmoys [5]. Our algorithm is based on local search and we analyze its performance by appealing to a packing bound of Hurkens and Schrijver [7]. We also show that PLSE is APX-hard, thereby strengthening the NP-hardness result of Colbourn [2].

We then study the k-PLSE problem. For this problem, we first show a natural greedy algorithm that achieves an approximation factor $\frac{1}{2}$. We also show that a randomized rounding procedure applied on the LP formulation of the problem achieves a factor of $1 - \frac{1}{e} - \epsilon$. Moreover we show that this is almost the best possible, i.e. no polynomial-time algorithm for k-PLSE can achieve factor better than $1 - \frac{1}{e} + \epsilon$ unless P = NP.

Finally, for the c-PLSE problem, based on a theorem of Ryser [13], we present a $\frac{1}{4}$-approximation algorithm.

2 Preliminaries

Let $[n] = \{1, \ldots, n\}$. A *partial latin square* (PLS) of order n is an $n \times n$ array whose cells are empty or contain a color from $[n]$. with the restriction that no color is repeated in a row or column. When the PLS has no empty cells it is simply called a *latin square* (LS). We denote the content of the (i, j)-th cell in the PLS L by $L(i, j)$. The number of non-empty cells of L is denoted $|L|$.

A PLS L' is an *extension* of a PLS L if $L'(i, j) = L(i, j)$ holds for all non-empty cells $L(i, j)$; we denote this by $L \preceq L'$. Naturally, L' can be obtained by coloring some of the empty cells in L.

The *partial latin square extension* problem (PLSE) is, given a PLS L, color the maximum number of empty cells in L using colors in $[n]$, i.e., find L' such that $L' \succeq L$ and $|L'|$ is maximized. The *k-partial latin square extension* problem (k-PLSE) is, given a PLS L of order n, color the maximum number of empty cells in L by colors in $[n]$ such that at most k colors are used in the coloring. It is clear that PLSE is the same as n-PLSE. The *c-partial latin square extension* problem (c-PLSE) is, given a PLS L of order n with T filled cells, find the largest $c, 0 \leq c \leq 1$, such that L contains a PLS with cT filled cells that can be extended to completion. It is clear that when an instance of c-PLSE has $c = 1$ it means that it can be extended to completion.

A ρ-approximation to these problems is to find a PLS L' such that $|L'|$ is within ρ of the optimum solution to the problem, where $0 < \rho \leq 1$.

2.1 The 3EDM problem

To facilitate the presentation of our results, we define the following new problem called *3EDM*: given a tripartite graph, this problem corresponds to finding the largest number of edge disjoint triangles in the graph. Similarly, in the *k-3EDM* problem, the goal is to find the largest number of edge disjoint triangles in a tripartite graph, with the constraint that at most k vertices from the third partition are touched by the triangles. We argue that 3EDM and PLSE problems are equivalent, i.e., there are value-preserving reductions from PLSE to 3EDM and vice versa.

Theorem 1. *The PLSE and 3EDM problems are equivalent.*

Proof. The reduction from PLSE to 3EDM is straightforward. Given an $n \times n$ instance L of PLSE, create a tripartite graph G with $3 \times n$ vertices as follows. The first partition in G represents the n rows, the second represents the n columns, and the third represents the n colors. For each empty cell (i, j) in L and each candidate color k that can be assigned to this cell, place a triangle between the vertices i, j, k in G. It is easy to see that L can be extended to t additional cells if and only if G has t edge-disjoint triangles.

Conversely, we show that there is a value-preserving reduction from 3EDM to PLSE. Let $G = (U \cup V \cup W, E)$ be a tripartite graph and let $n = \max\{|U|, |V|, |W|\}$. We construct a PLS L of order $3n$ such that maximum number of edge disjoint triangles in G equals the maximum number of entries that can be filled in L and vice versa.

First we assume that every edge in G is contained in at least one triangle, since edges that are not present in at least one triangle can always be removed without affecting the solution. Next we assume that $|V| = |U| = |W| = n$, since isolated vertices can be added to G without changing the solution. Let $U = \{u_1, \ldots, u_n\}, V = \{v_1, \ldots, v_n\}$, and $W = \{w_1, \ldots, w_n\}$. Let L be an empty PLS of order $3n$; think of L as being composed of square blocks $A_1, \ldots, A_9$, each of dimension $n \times n$; here the blocks are numbered in the row-major order. Now we turn L into a PLS such that the entry (i, j) in L is empty and can be filled with color $k \leq n$ if the triangle (u_i, v_j, w_k) exists in G.

Let R_i be the index set of vertices in W such that u_i is not connected to them. For each $r \in R_i$, we fill an empty entry in the i-th row of A_2 with color r. Note that we can do this for all $i = 1, \ldots, n$ without creating a conflict. Similarly let C_j be the index set of vertices in W such that v_j is not connected to them. For each $c \in C_j$, we fill one of the empty entries in the j-th column of A_4 with c. Now it is easy to see that we can fill the entry (i, j) in L with color $k \leq n$ if the triangle (u_i, v_j, w_k) appears in G. However it is possible to fill these entries with colors greater than n. To circumvent this problem, we use the additional blocks in the following way.

Let A_1' be the subset of entries in A_1 such that $(i, j) \in A_1'$ if the edge (u_i, v_j) does not appear in G. We fill in the entries in A_1' with colors from the set $\{n+1, \dots, 2n\}$; this will ensure that the non-edge (u_i, v_j) does not contribute to the PLSE solution. After this step, let A_{1j} be the set of colors appearing in j-th column of A_1. For every $r \in \{n+1, \dots, 2n\}$, if $r \notin A_{1j}$, then we place r in an empty entry in the j-th column of A_7. This way, we ensure that none of the colors in $\{n+1, \dots, 2n\}$ can be used to fill the empty entries of A_1. Analogously, the block A_3 is used to ensure that none of the colors in $\{2n+1, \dots, 3n\}$ can be used to fill the empty entries of A_1; this can be easily achieved by setting A_3 to be a complete latin square with entries from $\{2n+1, \dots, 3n\}$. Now it suffices to fill in the remaining entries greedily except that we have to avoid filling the entries of A_2, A_4, and A_7 with colors from $\{1, \dots, n\}$. We can block A_2 and A_4 w.r.t $\{1, \dots, n\}$ by placing appropriate colors in A_6 and A_8. We fill the empty entries in A_7 with colors from the set $\{2n+1, \dots, 3n\}$). Now all entries except the empty ones in A_1 are either filled or blocked and we can place k in (i, j) if and only if the triangle (u_i, v_j, w_k) exists in G. This completes the proof.

In a similar manner, we can show that

Corollary 1. *The k-PLSE and k-3EDM problems are equivalent.*

3 Improved bounds for the PLSE problem

In this section we obtain a $\frac{2}{3}$-approximation algorithm for the PLSE problem; this improves the $1 - \frac{1}{e}$ algorithm of Gomes, Regis, and Shmoys [5]. We then show that the PLSE problem is APX-hard.

3.1 A local search algorithm

First, we state a well-known result of Hurkens and Schrijver [7].

Theorem 2 (Hurkens–Schrijver Theorem [7]). *Let m, n, k, t be positive integers with $k \geq 3$. Let $E_1, \dots, E_m$ be subsets of a set V of size n such that*

1. *each element of V is contained in at most k of the sets $E_1, \dots E_m$ and*
2. *any collection of at most t sets among $E_1, \dots E_m$ has a system of distinct representatives.*

Then we have $\frac{m}{n} \leq \frac{k(k-1)^r - k}{2(k-1)^r - k}$ if $t = 2r - 1$ and $\frac{m}{n} \leq \frac{k(k-1)^r - 2}{2(k-1)^r - 2}$ if $t = 2r$.

We present a simple local search-based approximation algorithm for PLSE by obtaining an algorithm for 3EDM.

Theorem 3. *For any $\epsilon \geq 0$, there is a $\frac{2}{3} - \epsilon$-approximation algorithm for the 3EDM problem.*

Proof. Let G be the given graph with n vertices. Fix a $t \geq 1$. Start with any collection of edge-disjoint triangles from G. Iteratively perform local search by replacing any sub-collection of $s \leq t$ triangles with $s+1$ triangles from the graph such that the collection continues to be edge disjoint.

It is obvious that the above heuristic run in polynomial time since the collection grows by at least 1 in each step and its size is upper bounded by n^2. Let OPT denote the largest collection of edge disjoint triangles in G.

Now, we apply Theorem 2 to our situation by taking the sets $E_1, \ldots, E_m$ to be the edge disjoint triangles of OPT and V to be the collection of edge disjoint triangles found by our heuristic with edge intersection representing containment, i.e., we say E_i contains v_j, an element of V, when the intersection of the triangle in OPT corresponding to E_i with the triangle corresponding to v_j contains at least an edge of the original graph.

Observe that both the conditions of Theorem 2 are met:

1. since each of the two collections of triangles, the set corresponding to $E_1, \ldots, E_m$ as well as the set corresponding to V are edge disjoint therefore it follows that each E_i can intersect at most 3 v_j and vice versa and
2. by the termination condition of the heuristic every collection of t elements from $E_1, \ldots, E_m$ must have a system of distinct representatives in V, i.e., intersect at least t triangles from V for otherwise we could replace $s \leq t$ triangles from V with at least $s+1$ triangles from $E_1, \ldots E_m$.

Hence, when the heuristic terminates, the size of the collection as a fraction of $|\text{OPT}|$ is at least $(2 - \frac{3}{2^r})/(3 - \frac{3}{2^r})$ if $t = 2r - 1$ and $(2 - \frac{2}{2^r})/(3 - \frac{2}{2^r})$ if $t = 2r$. The proof is complete.

Note that the running time of the heuristic increases the closer we wish to get to $\frac{2}{3}$. In particular to beat the existing bound of $1 - \frac{1}{e}$ [5], we can run the heuristic with any $t \geq 7$. Naively implemented, the running time of the heuristic in this case is $O(n^{26})$ since we are picking upto 8 triangles at a time from a maximum possible collection of $O(n^3)$ triangles upto $O(n^2)$ times.

Corollary 2. *For any $\epsilon > 0$ there exists a polynomial time algorithm that approximates PLSE to within $\frac{2}{3} - \epsilon$.*

3.2 APX-hardness

In this section we show that 3EDM is APX-hard. We prove that in the reduction of Holyer [6], if we restrict the input 3SAT instances to the instances of 5-OCC-MAX-3SAT—each variable occurs exactly five times in the formula—then the reduction becomes gap preserving. Feige [4] proved that there is a constant ϵ such that it is not possible to distinguish between satisfiable instances of 5-OCC-MAX-3SAT and ones where at most ϵ fraction of clauses are satisfiable, unless

P = NP. (Holyer's reduction was also used in [8] to prove the APX-hardness of a variant of cycle covering. For sake of completeness here we repeat the definitions (to avoid confusion we use the notation used in [6, 8]).)

Theorem 4. *3EDM is APX-hard.*

Proof. Let the graph $H_{3,p}$ be a graph with p^2 vertices where $V = \{(x_1 + x_2 + x_3) \in Z_n^3 \mid x_1 + x_2 + x_3 \equiv 0(\text{mod } p)\}$ and two vertices $(x_1, x_2, x_3), (y_1, y_2, y_3)$ are connected if there are distinct i, j, and k such that $x_i \equiv y_i(\text{mod } p), x_j \equiv y_j + 1(\text{mod } p)$ and $x_k \equiv y_k - 1(\text{mod } p)$. As has been pointed out in [2], if we choose p so that $p \equiv 0(\text{mod } 3)$ the graph becomes tripartite. The crucial point is that there is just two ways to partition $H_{3,p}$ into triangles; this will serve as a switch for modeling a truth assignment. We call one partitioning a T-partition and the other an F-partition. We define a *patch* to be an induced subgraph in $H_{3,p}$ that consists of a triangle in center with three other triangles surrounding it. When the central triangle belongs to a T-partition, we call it a T-patch and otherwise an F-patch.

Let $\aleph$ be an instance of 5-OCC-MAX-3SAT that consists of $m = 5/3n$ clauses $C = (C_1, \ldots, C_m)$ defined over n variables $x_1, \ldots, x_n$ where each C_j consists of three literals $\ell_{j,1}, \ell_{j,2}$, and $\ell_{j,3}$. For each variable x_i in $\aleph$ we create a graph X_i that is a copy of $H_{3,6}$. Also corresponding to each literal $\ell_{j,k}$ we create a graph $C_{j,k}$ that is a copy of $H_{3,6}$. Now we glue the graphs in the following way. If $\ell_{j,k} = x_i$, then we glue an F-patch of X_i with an F-patch of $C_{j,k}$ and otherwise (when $\ell_{j,k} = \bar{x}_i$) we glue an F-patch of X_i with a T-patch of $C_{j,k}$. We also glue $C_{j,1}, C_{j,2}$, and $C_{j,3}$ together at an F-patch from them and then remove the edge of the central triangle in the F-patch. Note that we have chosen $p = 6$ so that we have enough disjoint number of patches, also to ensure that the resulted graph is tripartite, we glue the vertices with the same color. Let G_j be the graph after gluing together the graphs $C_{j,1}, C_{j,2}$, and $C_{j,3}$. The following facts have been shown in [6].

1. In order to partition all of the edges in G_j, exactly one of the graphs $C_{j,1}, C_{j,2}$, and $C_{j,3}$ should be F-partitioned.
2. If $\ell_{j,1} = x_i$, then it is not possible that $C_{j,k}$ and X_i are both F-partitioned. If $\ell_{j,k} = \bar{x}_i$ then it is not possible that $C_{j,k}$ is F-partitioned and X_i is T-partitioned.

These facts imply

Lemma 1. *The edges of the graph G_j can be partitioned into triangles if and only if one of the literals in C_j is true.*

Let t_1 be the number of edge-disjoint triangles in $H_{3,6}$ and let t_2 be the number of edge-disjoint triangles in G_j. Lemma 1 indicates that if $\aleph$ is satisfiable, then there are $nt_1 + 5/3nt_2 = c_1 n$ edge-disjoint triangles in the final graph, where c_1 is a constant. On the other hand if $\aleph$ is not satisfiable, then for each unsatisfiable clause C_j we have two possibilities: there is one edge that has been left or the there is one edge left in one of graphs corresponding to the variables involving in C_j. Since each variable is in at most two unsatisfied clause (otherwise we can switch it), we can conclude that if there are $(1-\epsilon)5/3n$ unsatisfiable clauses in $\aleph$, then we can have at most $t_1 n + 5/3 t_2 n - 5/6(1-\epsilon)n = c_2 n$ edge-disjoint triangles, where $c_2 < c_1$. This shows that there is a constant $\alpha < c_2/c_1$ such that if we can α-approximate the number of edge-disjoint triangles in tripartite graphs, then we can distinguish between satisfiable instances of 5-OCC-MAX-3SAT and instances that at most ϵ fraction of them are satisfiable. This completes the proof.

Corollary 3. *PLSE is APX-hard.*

4 The *k*-PLSE problem

In this section we study the k-PLSE problem. First we present a simple greedy algorithm that approximates to within a factor of $\frac{1}{2}$. Next we show a randomized approximation algorithm that achieves a factor $1 - \frac{1}{e} - \epsilon$. Finally, we prove that k-PLSE is hard to approximate to within a factor of $1 - \frac{1}{e} + \epsilon$.

4.1 A greedy algorithm

Let M_i be the largest matching that extends color i. Pick color j such that $|M_j| = \max\{|M_1|, \ldots, |M_n|\}$, breaking ties arbitrarily. Fill the cells in M_j with color j and repeat until k colors are used.

Theorem 5. *The greedy algorithm approximates k-PLSE to within a factor $\frac{1}{2}$.*

Proof. Let OPT $= \{M_{x_1}, \ldots, M_{x_k}\}$ be the optimum solution where M_{x_i} is the set of cells that have received color x_i. Accordingly let $S = \{M'_{y_1}, \ldots, M'_{y_k}\}$ be the solution produced by the greedy algorithm. For each cell $(i,j) \in$ OPT we determine a cell $(i',j') \in S$ as accountable. If $(i,j) \in S \cap$ OPT, then (i,j) is accountable for itself, otherwise we distinguish two cases. Suppose $(i,j) \in M_{x_c}$. First case: x_c has been used in S. In this case, we can determine a cell (i',j') such that $(i',j') \in M'_{x_c}$ and $i' = i$ or $j = j'$. Note that this can make a cell accountable to two cells but not more. Second case: x_c has not been used in S. Let t_{x_c} be the number of cells in OPT with color x_c such that these cells are left unfilled in S. For each unused color x_c in OPT we can determine a color $y_{c'}$ in S such that $y_{c'}$ does not appear in OPT. For each $1 \leq c \leq k$, $|M'_{y_c}| \geq t_{x_c}$ for otherwise x_c should have been selected in the algorithm. Therefore, for each unfilled cell in M_{x_c} we can determine a cell in $M'_{y_{c'}}$ as accountable. The cells that have been chosen as accountable are at most accountable to one cell (itself). Hence in overall each cell is accountable to at most two cells in the optimum solution and this completes the proof.

Let $k = 3$ and consider the PLS

	2	3	
2			1
3			2
	1	2	

. The greedy algorithm first chooses color 4 and if it decides to color the main diagonal. By this choice, at most 4 cells can be filled while the optimum solution can be shown to color 8 cells. This example shows that the above analysis is tight.

4.2 A $1-\frac{1}{e}-\epsilon$ approximation algorithm

In this section we modify the LP formulation for PLSE problem defined in [5] to get a $(1-\frac{1}{e}-\epsilon)$-approximation for the k-PLSE problem. Let $\mathcal{M}_c$ be the set of all matchings that extend the matching associated with color c and y_{cM} be the indicator variable associated with matching $M \in \mathcal{M}_c$. The modified formulation is:

$$\begin{aligned}
\text{maximize} \quad & \sum_{c=1}^{n} \sum_{M\in\mathcal{M}_c} |M| y_{cM} && (1)\\
\text{subject to} \quad & \forall c = 1,\ldots,n: \sum_{M\in\mathcal{M}_c} y_{cM} = 1 \\
& \forall i,j = 1,\ldots,n: \sum_{c=1}^{n} \sum_{M\in\mathcal{M}_c:(i,j)\in M} y_{cM} \le 1 \\
& \sum_{c=1}^{n} \sum_{M\in\mathcal{M}_c:|M|>0} y_{cM} \le k \\
& y_{cM} \ge 0
\end{aligned}$$

We follow the same rounding technique used in [5] except that before applying the rounding procedure on the LP solution, we multiply each variable in the solution by $1-\epsilon$. Given that, we can use the Chernoff bound to guarantee that at most k matchings with different colors have been picked with some constant probability.

Theorem 6. *Let $k \ge \frac{2}{\epsilon^2}(1-\epsilon)(\ln\frac{1}{\delta})$, $0 < \epsilon \le \frac{1}{2}$, and $0 \le \delta \le 1$. There is a randomized $(1-\frac{1}{e}-\epsilon)$-approximation algorithm for k-PLSE that succeeds with probability at least $1-\delta$.*

Proof. Let y^* be the optimal solution for the above LP and $\bar{y}$ be the solution obtained from y^* after multiplying each variable by $1-\epsilon$. Now for each color c, we pick a matching from the set $\mathcal{M}_c$ such that matching y_{cM} is picked with probability $\bar{y}_{cM}$. If two or more mathcings share cell (i,j) we color (i,j) arbitrary with the color of one those matchings. Let OPT and OPT' be the cost of $y*$ and $\bar{y}$ respectively. According to the argument in [5], the cost of solution produced by the above rounding procedure is at least $(1-\frac{1}{e})\text{OPT}'$. Since $\text{OPT}' = (1-\epsilon)\text{OPT}$, we conclude that the cost of final solution is at least $(1-\epsilon)(1-\frac{1}{e})\text{OPT} \ge (1-\frac{1}{e}-\epsilon)\text{OPT}$. It remains to prove that the solution is feasible, i.e., at most k different colors have been picked. Let $s = \sum_{c=1}^{n} \sum_{M\in\mathcal{M}_c:|M|>0} \bar{y}_{cM}$. We have $E(s) \le (1-\epsilon)k$ and since s is the sum of a set of independent random variables, we can apply the version of Chernoff bound used in [10] to bound the tail of s. Given $0 \leq \epsilon' \leq 1$ such that $\epsilon' = \frac{\epsilon}{1-\epsilon}$, we have,

$$\Pr[s > k] = \Pr[s > (1+\epsilon')(1-\epsilon)k] \leq \exp\left(-\frac{(1+\epsilon)(1+\epsilon')^2 k}{2}\right) \leq \delta.$$

After simplification, we have $\Pr[s > k] < \delta$ when $k \geq \frac{2}{\epsilon^2}(1-\epsilon)(\ln \frac{1}{\delta})$. This completes the proof.

Note that if we settle for some constant probability of success, we can use brute-force search for values of k less than $\frac{2}{\epsilon^2}(1-\epsilon)(\ln \frac{1}{\delta})$.

4.3 Hardness

We show that the k-PLSE problem is hard to approximate to within $1 - \frac{1}{e} + \epsilon$, unless P = NP.

Theorem 7. *For any $\epsilon > 0$, k-3EDM is not approximable to within $1 - 1/e + \epsilon$, unless P = NP.*

Proof. We use the Max-k-Cover problem for the reduction. In the *Max-k-Cover* problem, we are given several subsets of a ground set and we are asked to pick k subsets that cover most of the ground set elements. Feige [4] proved that no polynomial time algorithm for Max-k-Cover can have approximation ratio better than $1 - \frac{1}{e}$, unless P = NP.

Given an instance of Max-k-Cover with the ground set $\{e_1, \ldots, e_n\}$ and subsets $S_1, \ldots, S_m$, we construct the tripartite graph $G = (U \cup V \cup W, E)$ in the following way. Let $|U| = |V| = n$ and $|W| = m$. We place a perfect matching between U and V where the edge (u_i, v_i) correspond to the element e_i. Now if $e_i \in S_k$ we connect u_i and v_i to w_k, thereby creating the triangle (u_i, v_i, w_k). It is easy to see that every solution to the given instance of Max-k-Cover problem corresponds to a solution to the k-3EDM problem and vice versa.

Corollary 4. *For any $\epsilon > 0$, k-PLSE is not approximable to within $1 - 1/e + \epsilon$, unless P = NP.*

5 The c-PLSE problem

In this section we present a $\frac{1}{4}$ approximation algorithm for the c-PLSE problem.

Theorem 8. *There exists a polynomial-time algorithm that approximates the c-PLSE problem to within a factor $\frac{1}{4}$.*

Proof. In fact we show a stronger result, namely that: every partial Latin square with T filled cells has a subset with size of at least $T/4$ filled cells that can be extended to completion.

Let P be a partial Latin square of order n with t filled cells. We distinguish two cases. $n = 2m$: we divide the square into four blocks of size $m \times m$ and then pick the block that have more filled cells ($\geq t/4$). By permuting rows and columns, we exchange the picked block with the left upper hand block and then clear the other cells. It is easy to see that we can complete the upper-left block in any order. And for completing the square we invoke a famous theorem of Ryser ([11], also [13, Theorem 17.4]) that we state for the sake of completeness:

Theorem 9 (Ryser's Theorem [11]). *Let A be a partial Latin square of order n in which cell (i, j) is filled if and only if $i \leq r$ and $j \leq s$ Then A can be completed if and only if $N(i) \geq r + s - n$ for $i = 1, \dots, n$, where $N(i)$ denotes the number of elements of A that are equal to i.*

By Ryser's theorem (letting $r = s = m$), the square is guaranteed to be extendable to completion.

The proof for the situation when $n = 2m + 1$ is similar except that here we divide the square into four blocks of size $m \times (m - 1)$ with one cell left in the center of square. If the cell which is left in the center is not empty, we permute the rows and columns so that it becomes an empty cell. Again we pick the block with more filled cells and using the above lemma (let $r = m, s = m + 1$), we are done.

We make the following remarks on the above theorem. Using the above approach it is not possible to get better than a $\frac{1}{2}$ approximation. Consider a square of order 2n. Put a LS from the colors $\{1, \dots, n\}$ in the upper left section of the square and similarly, put a LS from colors $\{n + 1, \dots, 2n\}$ in the bottom-right section. It is easy to see that this is a blocked PLS and moroever in order to obtain a completable subset of the filled cells, we have to cancel at least $\frac{n^2}{4}$ of the filled cells (In order to put a number in the empty cells at least one filled cell should be canceled). In fact the combinatorial version of the c-PLSE problem is in of itself a very interesting problem — what is the largest fraction f, $0 < f < 1$ such that every PLS with T filled cells contains a PLS with at least fT filled cells that can be extended to completion? We conjecture that the right answer is $f = \frac{1}{2}$.

6 Conclusions

We defined two new and natural problems - k-PLSE and c-PLSE. We obtained simple approximation algorithms for the PLSE, k-PLSE and c-PLSE problems. We also showed APX-hardness for PLSE and a $(1-\frac{1}{e})$-hardness of approximation for k-PLSE. Our result for PLSE is an improvement over the best known and our result for k-PLSE is the best possible.

The main open problem is to improve the approximation ratio for PLSE. Obtaining an *explicit* constant hardness of approximation is also an interesting problem. Although there is a $(1-\frac{1}{e})$ hardness result for k-PLSE, the improvement for approximation of PLSE is not unlikely as the hardness of k-PLSE seems to be of a different origin—for example, the worst-case instance for k-PLSE is an easy instance for PLSE.

Embedding PLSs in LSs with the same order and with minimum loss of elements poses many new directions and open problems. We conjecture that the tight constant in Theorem 8 is $\frac{1}{2}$.

References

1. R. A. Barry and P. A. Humblet. Latin routers, design and implementation. *IEEE/OSA Journal of Lightwave Technology*, pages 891–899, 1993.
2. C. J. Colbourn. The complexity of completing partial latin squares. *Discrete Applied Mathematics*, 8:25–30, 1984.
3. T. Evans. Embedding incomplete latin squares. *American Mathematical Monthly*, 67:958–961, 1960.
4. U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.
5. C. P. Gomes, R. G. Regis, and D. B. Shmoys. An improved approximation algorithm for the partial latin square extension problem. *Oper. Res. Lett.*, 32(5):479–484, 2004.
6. I. Holyer. The NP-completeness of some edge-partition problems. *SIAM J. Comput.*, 10(4):713–717, 1981.
7. C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discret. Math.*, 2(1):68–72, 1989.
8. N. Immorlica, M. Mahdian, and V. S. Mirrokni. Cycle cover with short cycles. In *STACS*, pages 641–653, 2005.
9. R. Kumar, A. Russell, and R. Sundaram. Approximating latin square extensions. *Algorithmica*, 24(2):128–138, 1999.
10. P. Raghavan, C.D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica* 7(4): 365-374, 1987.
11. H. J. Ryser. A combinatorial theorem with an application to latin rectangles. *Proc. Amer. Math. Soc.*, 2:550–552, 1951.
12. B. Smetaniuk. A new construction on Latin squares I. A proof of the Evans conjecture. *Ars Combinatoria*, XI:155–172, 1981.
13. J. H. van Lint and R. M. Wilson. *A Course in Combinatorics*. Cambridge University Press, 1992.