List coloring of Latin and Sudoku graphs

Antal Iványi and Zsolt Németh
Department of Computer Algebra of Eötvös Loránd University
tony@compalg.inf.elte.hu
Department of Numerical Analysis of Eötvös Loránd University
birka0@gmail.com

Abstract

In 2001 Galvin [44] proved a generalization of Hall theorem [49]. We improve his result in two directions, giving a weaker precondition and also a stronger consequence. We show some basic properties of the Latin and Sudoku graphs considering the solution of the Latin and Sudoku puzzles as a list coloring problem of graphs [5, 11, 14, 20, 33, 39, 57, 58, 60, 62, 84].

1 Introduction

List coloring of graphs was proposed by Vizing in 1976 [87]. Substantial progress was made by Erdős, Rubin and Taylor in 1979 [31].

Today graph coloring is a popular research topic of combinatorics and computer science. Concrete applications appear in connection with VLSI design [30, 61], networks [27, 38, 66, 73], resource allocation [45], and determination of partially given Latin [34] and Sudoku [17, 18, 19, 37, 40, 63, 64, 65] squares.

Let $m \geq 1$ and $n \geq 2$ be integers and $G = (V, E)$ be a finite simple graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set $E = \{e_1, e_2, \ldots, e_m\}$. Assume further that an infinite set $C = \{c_1, c_2, \ldots\}$ of so called colors, and a collection $L = \{L(v_1), L(v_2), \ldots, L(v_n)\}$ of lists of colors are given, where $L(v_i) \subseteq C$ for $1 \leq i \leq n$. Here each list $L(v_i)$ is considered as the set of allowed colors for vertex $v_i$. $L$ is called list arrangement.

For the simplicity and without the loss the generality we suppose that $C$ is the set of positive integers, and so each list is a finite set of positive integers. If the sizes of the lists have a join upper bound, that is if $|L(v_i)| \leq k$, then the collection $L$ is called a $k$-assignment on $G$. Since the components of $G$ are colorable independently, for the simplicity we suppose that $G$ is connected.

A sequence of colors (positive integers) $S = (c_{i_1}, c_{i_2}, \ldots, c_{i_n}) = (i_1, i_2, \ldots, i_n)$ is called a proper vertex coloring of $G$ with a given collection of lists $L$, if for all $1 \leq j < k \leq n$

1. $i_j \in L(v_j)$;
2. if $v_j v_k \in E$, then $i_j \neq i_k$.

Graph $G$ with a given $L$ is $L$-colorable (list colorable), if there exists a proper vertex coloring function $\varphi : V \to C$ satisfying for all for all $1 \leq j < k \leq n$

1. $\varphi(v_j) \in L(v_j)$;
2. if $v_j v_k \in E$, then $\varphi(v_j) \neq \varphi(v_k)$.

Figure 1 shows a graph $G_1$ with some list assignment $A_1$.

Graph $G_1$ with list assignment $A_1$ has two proper colorings $S_1$ and $S_2$, which are represented in Figure 2.

Figure 3 shows graph $G_2$ with list assignment $A_2$, but this graph has no proper list coloring. Let $f$ a function from $V$ to the positive integers, that is

$$f : V \to \{1, 2, \ldots\}.$$
then a graph $G$ is called $f$-choosable [31] if there is a proper $\mathcal{L}$-coloring of $G$ whenever $\mathcal{L}$ satisfies

$$|\mathcal{L}| \geq f(v_i) \text{ for all } v_i \in V.$$ 

For example the graph $G_1$ represented in Figure 1 is not $f$-choosable, if $f$ is defined according to Figure 4. In Figure 5 such list assignment $A_3$ is represented which shows that the graph $G_1$ is not $f$-choosable.

The most important graph parameter in list coloring is the choice number of a graph $G$ denoted by $\chi_l(G)$ and defined as the smallest integer $k$ such that $G$ is $\mathcal{L}$-colorable for every $k$-assignment $\mathcal{L}$.

The choice number of the graphs represented in Figures 1 and 3 is 3. The $\chi(G)$ chromatic number of these graphs is also 3. It is known that $\chi(G) \leq \chi_l(G)$ for all $G$ [31, 87].

The exact characterization of the graphs for which the chromatic number is equal with the choice number is an open problem. The graph $G_4$ represented in 6 represented with a list assignment $A_4$ shows $\chi_l(G_4) = 3$, while $\chi(G_4) = 2$. 
Figure 4: Function $f_1$ for graph $G_1$.

Figure 5: List assignment $A_3$ showing that graph $G_1$ is not $f_1$-choosable.

Figure 6: Graph $G_4$ with list assignment $A_4$.

2 Extension of Hall theorem

The Hall theorem [49] results the following necessary and sufficient condition of the $L$-colorability of $K_n$ (the complete graph on $n$ vertices): $K_n$ is $L$-colorable if and only if the union of any $k$ ($1 \leq k \leq n$) lists contain at least $k$ elements.

The classical Hall theorem has several new proofs [7, 8, 12, 71, 72, 76, 82] and many extensions [1, 2, 3, 4, 13, 25, 24, 26, 31, 35, 41, 42, 47, 48, 50, 52, 54, 55, 56, 69, 70, 74, 89].


Let $G$ be a finite, undirected, simple graph with vertex set $V$. Let $C = \{C_x : x \in V\}$ be a family of sets indexed by the vertices of $G$. For $X \subseteq V$, let $C_X = \cup_{x \in X} C_x$. A set $X \subseteq V$ is $C$-colorable if one can assign to each vertex $x \in X$ a "color" $c_x \in C_x$ so that $c_x \neq c_y$ whenever $x$ and $y$ are adjacent in $G$. 
a) Prove that if \(|C_X| \geq |X|\) whenever \(X\) induces a connected subgraph of \(G\), then \(V\) is \(C\)-colorable.

b) Prove that if every proper subset of \(V\) is \(C\)-colorable and \(|C_V| \geq |V|\), then \(V\) is \(C\)-colorable.

c) For every connected graph \(G\), find a family \(C = \{C_x : x \in V\}\) showing that the condition \(|C_V| \geq |V|\) in part b) cannot be weakened to \(|C_V| \geq |V| - 1\).

Part a) was solved by Stephen C. Locke (Florida Atlantic University, Boca Raton, FL), part b) by Sung Soo Kim (Hanyang University, Ansan, Kyunggi, Korea), and part c) by David Callan (University of Wisconsin, Madison, WI) in 2001 [44, 67].

These assertions on list coloring of graphs can be strengthened in two directions:

• there is a weaker sufficient condition than in part a);

• the condition in part a) implies a stronger consequence for which the given condition is necessary and sufficient.

Let \(\Gamma(x) = \{y : x \text{ and } y \text{ are adjacent in } G\}\) and \(\text{deg}(x) = |\Gamma(x)|\). A set \(X \subseteq V\) is \(D\)-colorable if one can assign to each vertex \(x \in X\) a color \(c_x \in C_x\) so that \(c_x \neq c_y\) whenever \(x\) and \(y\) are connected with a path in \(G\).

d) Prove that if \(\text{deg}(v) > |C_v|\) for each \(v \in V\), then \(G\) is \(C\)-colorable.

e) Prove that \(G\) is \(D\)-colorable iff \(|C_X| \geq |X|\) whenever \(X\) induces a connected subgraph of \(G\).

Solution to part d)

Let \(V = (v_1, v_2, \ldots, v_{|V|})\). Then we can assign colors to vertices e.g. in the increasing order of their indices, since a preceding vertex can exclude at most the colors of \(\Gamma(v_i)\) adjacent vertices, but the degree of \(v_i\) is greater than the number of its adjacent vertices.

Solution to part e)

If \(G\) is \(D\)-colorable and \(X\) induces a connected subgraph of \(G\), then the elements of \(X\) are connected with a path in \(G\), therefore \(|C_X| \geq |X|\) holds due to the definition of \(D\)-colorability.

If \(|C_X| \geq |X|\) whenever \(X\) induces a connected subgraph \(S_X\) of \(G\), then \(S_X\) is \(D\)-colorable due to Hall’s theorem and \(D\)-colorability of the connected subgraphs of \(G\) implies the \(D\)-colorability of \(G\).

We remark that if \(G\) is connected, then part e) is equivalent with Hall’s theorem [49].

3 List coloring of Latin and Sudoku graphs

For a positive integer \(n \geq 2\) the \(n\)-order Latin graph \(\lambda_n = (V_n, E_n)\) corresponds to an \(n\)-order Latin square [28, 29] and is the finite simple graph with \(V_n = \{v_{ij} : 1 \leq i, j \leq n\}\), and \(E_n = \{(v_{ab}, v_{cd}) : a = c \text{ and } b \neq d, \text{ or } b = d \text{ and } a \neq c\}\).

Figure 9 shows the \(n^2 = 16\) vertices of \(\lambda_4\). \(\lambda_4\) has \(n \times n \times (n - 1) = 48\) edges:

\[E_4 = \{v_{11}v_{12}, v_{11}v_{13}, v_{11}v_{14}, v_{12}v_{13}, v_{12}v_{14}, v_{13}v_{14}, \ldots, v_{43}v_{44}, v_{11}v_{21}, \ldots, v_{34}v_{44}\}.\]

![Figure 7: The vertices of \(\lambda_4\).](image-url)
For a positive integer $m \geq 2$ the $m$-order Sudoku graph $\sigma_m = (V_m, E_m)$ [51, 78, 90] corresponds to an $m$-order Sudoku square [17, 18, 19, 37, 40, 63, 64, 65], which is an $m^2$-order Latin square divided into $m \times m$ disjoint subsquares of size $m \times m$ called blocks. So $\sigma_m$ is the finite simple graph with $V_m = \{v_{ij} : 1 \leq i, j \leq m^2\}$, and $E_m = \{(v_{ab}, v_{cd}) : a = c$ and $b \neq d,$ or $b = d$ and $a \neq c$ or $|(a + 1)/m| = [(c + 1)/m],$ and $|(b + 1)/m| = [(d + 1)/m]$ and $(a, b) \neq (c, d)$.  

Figure 8 shows the $9^2 = 81$ vertices of $\sigma_3$, $\sigma_3$ has $m^2 \times 2^2 = (m^2 - 1)$ = 648 edges in the rows and columns, plus $m^2 \times m^2 \times (m - 1)^2/2 = 162$ edges in the blocks, so $|E_3| = 810$ as follows:

$$E_3 = \{v_{11}v_{12}, \ldots, v_{18}v_{19}, \ldots, v_{91}v_{92}, \ldots, v_{98}v_{99},$$

$$v_{11}v_{21}, \ldots, v_{81}v_{91}, \ldots, v_{19}v_{29}, \ldots, v_{89}v_{99},$$

$$v_{11}v_{22}, \ldots, v_{23}v_{32}, \ldots, v_{77}v_{88}, \ldots, v_{89}v_{98}\}.$$  

![Figure 8: The vertices of $\sigma_3$.](image)

The following algorithm [28, 29]—defined using the pseudocode of [22]—produces a list coloring of an $n$-order Latin square $\lambda_n$ for the case $L(v_{ij}) = \{1, 2, \ldots, n\}$ for all $v_{ij}$. The input parameter is $n$, and the output is a matrix $M_\lambda = [m_{ij}]_{n \times n}$ of the colors of the vertices of the graph $\lambda_n$.

**LATIN($n, M_\lambda$)**

01 **for** $i \leftarrow 1$ **to** $n$

02 **do for** $j \leftarrow 1$ **to** $n$

03 **do** $M_\lambda[i, j] \leftarrow i + j - 1 \mod n$

04 **return** $M_\lambda$

The order of the running time of this algorithm corresponds to the number of elements of the matrix $M_\lambda$, it is $\Theta(n^2)$.

If the input is $n = 4$, then the output of LATIN is the coloring represented in Figure 9.

![Figure 9: An $\mathcal{L}$-coloring of $\lambda_4$ produced by algorithm LATIN.](image)

It is known [79] that $\lambda_4$ has 576 different $\mathcal{L}$-colorings, if all vertex $v_{ij}$ have the same color list $L(v_{ij}) = \{1, 2, 3, 4\}$, and $\lambda_3$ has $5,524,751,496,156,892,842,531,225,600 \sim 5.5 \cdot 10^{27}$ different colorings [79, 88], if all color lists are $L(v_{ij}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. These numbers give the number of 4-order, resp. 9-order Latin squares.
The following algorithm [83] produces a list coloring of the $m^2$-order Sudoku graph $\sigma_m$ for the case $L(v_{ij}) = \{1, 2, \ldots, m^2\}$ for all $v_{ij}$. Its basic idea is similar to the idea of algorithm LATIN. The input parameter is $m$, and the output is the matrix $M_\sigma = [m_{ij}]_{n \times n}$.

**SUDOKU**($m, M_\sigma$)

01 for $i \leftarrow 1$ to $m^2$
02 do for $j \leftarrow 1$ to $m^2$
03 do $M_\sigma[i, j] \leftarrow i + j - 1 + \lfloor (i - 1)/m \rfloor + m\lfloor (i - 1)/m \rfloor \mod m^2$
04 return $M_\sigma$

The running time of this algorithm also corresponds to the number of elements of the matrix $M_\sigma$: it is $\Theta(m^4)$.

If the input is $m = 3$, then the output of SUDOKU is the coloring represented in Figure 10.

![Figure 10: The $L$-coloring of $\sigma_3$, produced by algorithm SUDOKU.](image)

It is known [80] that $\sigma_2$ has 288 different $L$-colorings, if all lists in $L$ are equal to $\{1, 2, 3, 4\}$. It is also known [36, 77, 80, 85] that $\sigma_3$ has $667093752021072936960 \approx 6.7 \cdot 10^{21}$ different $L$-colorings, if all lists in $L$ are equal with $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. These numbers give the number of 4-order, resp. 9-order Sudoku squares.

In the typical Latin and Sudoku puzzles the color of some subset of vertices is fixed (these vertex-color pairs are the input data of the puzzle). The aim is to complete the coloring, that is determine the number of possible complete colorings and to determine the proper colorings.

The first complexity result is due to Colbourn who proved in 1984 [21], that the problem of decision whether the coloring of a partially colored Latin graph can be completed is NP-complete.

Using Colburn’s result Yato and Seta proved in 2003 [83], that the Sudoku ASP problem is NP-complete: if we know a proper vertex coloring of a partially colored Sudoku graph, then it is NP-complete problem to decide whether the given graph has another coloring or not.

In 1956 Behrens [9] proposed a more general problem, than the coloring of the Sudoku graph. He divided the set of the $n^2$ vertices of the $n$-order Latin graph $\lambda_n$ into $n$ disjunct subsets, and introduced the additional requirement that the elements of each subset have to be colored by different colors.

In 2007 Cameron asked [16] the complexity of the completion of the coloring of such generalized graphs, and Vaughan proved in 2009 [86] the NP-completeness.

Of course there are special graphs which are colorable by polynomial algorithms. E.g. the books written by Bach, Berthier, Erickson, Gordon, Inkala and Stuart [6, 10, 32, 46, 59, 81] and papers written by Crook [23] and Provan [75] contain numerous polynomial algorithms solving special Sudoku puzzles having unique solution.

**Acknowledgement.** The European Union and the European Social Fund have provided financial support to the project under the grant agreement no. TÁMOP 4.2.1/B-09/1/KMR-2010-0003.
References


