Two Genetic Algorithm for Ramsey Graphs

S. Masih Ayat1*, S. Majid Ayat1, Abbas Akrami1

1University of Zabol, Zabol, Sistan and Balouchestan, Iran

Abstract:
In this work, we use genetic algorithm (GA) for Ramsey graphs. Two versions of this algorithm are based on CPP and MATLAB.

Keywords: Genetic Algorithm; Ramsey Graphs

1. Introduction
In this paper, we only use simple undirected graphs: without multi-edges and loops. The Ramsey's theorem states that for any pair of positive integers s, t, there exists a least positive integer R(s, t) such that for any complete graph on R(s, t) vertices, whose edges are coloured red or blue, there exists either a complete subgraph on s vertices which all edges are blue, or a complete subgraph on t vertices which all edges are red.

Several algorithms are presented in Ramsey theory to produce two colorings of complete graphs without such complete subgraphs. These algorithms are only produce the monochromatic subgraphs of these two colorings. In fact, the adjacency matrices of these graphs are presented. In these matrices, the (i, j) entries are 1 if and only if the edge between \( v_i, v_j \) have the same color.

In this work, two genetic algorithms are presented which produce some iterations from parent graphs. These algorithms have different methods.

* Corresponding author. Email addresses. smasih@ipm.ir
2. Preliminaries

Definition 2-1 Let $G$ be a simple graph with $n$ vertices and $m$ edges. A $s$-clique in $G$ is a complete subgraph of $G$ with $s$ vertices. An independent set of order $t$, is a subset $\{v_1, \ldots, v_t\}$ vertices of $G$, without any edges between them in $G$. We call $G$ is a $(s, t, n)$-graph if $G$ has $n$ vertices, and without $s$-cliques and independent set of order $t$.

Definition 2-2 Let $G$ be a simple graph and $v \in V(G)$. Define $N(v) \subseteq V(G)$ the neighbor set of $v$, the set of vertices of $G$ such as $w$, which adjacent to $v$. If $W \subseteq V(G)$, then $G[W]$ is the induced subgraph of $G$ produced by $W$. In fact, $V(G[W]) = W$. Especially, define $G[N(v)]$ the neighbor graph of $v$.

Definition 2-3 Let $s$ be a positive integer number and $G$ be a $(s, s, n)$-graph. We call $G$ a diagonal Ramsey graphs, because the number of forbidden clique and forbidden independent sets are the same.

The following theorem is the base theorem of the first algorithm: The Gluing Algorithm. This theorem use the neighbor graphs.

Theorem 2-1 Let $G$ be a $(s, t, n, m)$-graph and $v \in V(G)$. Then $G[N(v)]$ is a $(s - 1, t, d(v))$-graph and $G[V(G) \setminus (N(v) \cup \{v\})]$ is $(s, t - 1, n - 1 - d(v))$-graph. Therefore,

$$R(s, t) \leq R(s - 1, t) + R(s, t - 1).$$

3. Algorithms

First, we present two main algorithms: The Gluing Algorithm and The Greedy Algorithms. The first algorithm find a $(s, t, n)$-graph if and only if $n < R(s, t)$, certainly. But the second algorithm find graphs with $n$ vertices such that the number of $s$-cliques and independent sets of order $t$ are decreasing by iterations. Both of them are algorithm suitable for arbitrary positive integers $s, t$. But if $s = t$, the usage of second algorithm is very simple than the other. If $s = 3$, the first is more better.

The gluing algorithm has graphical nature and has low speed. The second algorithm is a greedy algorithm and has high speed. Unfortunately the second find graphs which are approximately diagonal Ramsey graphs.

3-1 The Gluing Algorithm

This algorithm is well-defined by Theorem 2-1. We need a data base of all $(s - 1, t, n_1)$-graphs $G_1$ and $(s, t - 1, n_2)$-graphs $H_1$. Then set $v_0$ and the vertices of $G_1$ are the only neighbors of $v_0$. Now it is important how set edges between vertices of $G_1$ and $H_1$ to produce $(s, t, n_1 + n_2 + 1)$-graphs. This algorithm is applied by McKay and Radziszowski to prove that there exist no $(4, 5, 25)$-graph,[2]. Now, we present the generalization of this algorithm.
In fact, let $A_1, A_2$ be the adjacency graph of $G_1$ and $H_1$, respectively. Thus the adjacency graph of the product is as follows:

$$
\begin{bmatrix}
0 & 1 & \cdots & 1 & 0 & \cdots & 0 \\
1 & & & & & & \\
\vdots & A_1 & B & & & & \\
1 & & & & & & \\
0 & & & & & & \\
\vdots & B^T & A_2 & & & & \\
0 & & & & & & 
\end{bmatrix}
$$

The $B$ is $n_1 \times n_2$ matrix and is called the gluing matrix. Two approach could be present for finding this matrix.

**Case 1. Probabilistic Method**- We present a probabilistic algorithm to find an arbitrary $(0, 1)$-matrix of order $n_1 \times n_2$. This case is very easy to use, but very un-efficient algorithm. One could apply some changes randomly on 0s and 1s and hopes to find better gluing matrix!!

**Case 2. Graphical Method**- This method is very useful and sufficient but unfortunately very low speed!! This algorithm is as follows:

1. Enumerate the cliques and independent sets of $G_1$. The maximum cliques have size less than $s - 1$ and the maximum independent set have order less than $t$.
2. Enumerate the cliques and independent sets of $H_1$. The maximum cliques have size less than $s$ and the maximum independent set have order less than $t - 1$.
3. Suppose $1 \leq r \leq s - 1$. Let $\{w_1, \cdots, w_r\}$ be vertices of a $r$-clique in $G_1$ and $N_1, \cdots, N_r$ be the neighbors of $w_i$s in $H_1$, respectively. Then $N_1 \cap \cdots \cap N_r$ does not have $(s - r)$-cliques. Otherwise, this gluing is not suitable.
4. Suppose $1 \leq l \leq t - 1$. Let $\{w_1, \cdots, w_l\}$ be vertices of an independent set of order $l$ in $G_1$ and $N_1, \cdots, N_l$ be the neighbors of $w_i$s in $H_1$, respectively. Then $H_1 \setminus (U_{i=1}^l N_i)$ does not have any independent set of order $t - l$. Otherwise, this gluing is not suitable.
5. If there exist some gluing satisfy the condition 3 and 4 for all $r$-cliques and independent sets of order $l$ of $G_1$ for all $1 \leq r \leq s - 1$, $1 \leq l \leq t - 1$, then we have $(s, t, n)$-graphs. Otherwise, $n \geq R(s, t)$.

**3-2 The Greedy Algorithm**

This algorithm is suitable for arbitrary positive integers $s, t$. But we suppose $s = t$. We find the new adjacency matrices from their parents by some additional conditions: The new graphs have cliques and independent sets less than their parents.

We don't claim that this algorithm find $(s, t, n)$-graphs. This algorithm only get two graphs of order $n$ and generate a new graph such that this graph has cliques and independent set less than their parents!
This algorithm is as follows:
1. Let \( (0, 1) \)-matrices of order \( n \).
2. \( \text{CounterGr}(n, A, s) \) find all \( \{v_1, \ldots, v_s\} \) from \( A \), a \( (0, 1) \)-matrix of order \( n \) such that \( G[v_1, \ldots, v_s] \) are \( s \)-clique or independent set of order \( s \).
3. Give parents graphs \( Fa, Mo \) of order \( n \).
4. Procedure \( \text{FindMaxim}(Fa) \) find the maximum subgraph of \( Fa \) graph which has the clique number less than \( s \) and independent set number less than \( s \). This subgraph has vertex set \( \{w_1, \ldots, w_k\} \). One could rearrange the vertex set of \( Fa \) such that the first vertices of adjacency matrix of \( Fa \) are \( \{w_1, \ldots, w_k\} \). We set the adjacency matrix of \( \{w_1, \ldots, w_k\} \) by \( F \).
5. Procedure \( \text{FindMaxim}(Mo) \) find the maximum subgraph of \( Mo \) graph which has the clique number less than \( s \) and independent set number less than \( s \). This subgraph has vertex set \( \{u_1, \ldots, u_p\} \). One could rearrange the vertex set of \( Mo \) such that the last vertices of adjacency matrix of \( Mo \) are \( \{u_1, \ldots, u_p\} \). We set the adjacency matrix of \( \{u_1, \ldots, u_p\} \) by \( M \).
6. Procedure \( \text{KidMake}(Fa, Mo) \) set the \( \text{Kid} \) graph from parents \( Fa, Mo \) such that if \( k + p \geq n \), its vertex set is \( \{w_1, \ldots, w_k, u_1, \ldots, u_p\} \). Otherwise, the reminder vertices are choosen from \( Fa \). Set \( F_1 \) be the adjacency matrix of induced graph produced by these reminder vertices from \( Fa \), and \( F_2 \) be the adjacency part of \( Fa \) of vertices \( F \) versus \( F_1 \). The adjacency matrix of \( \text{Kid} \) is as follows:

\[
\begin{bmatrix}
F & B & F_2 \\
B^T & M & 0 \\
F_2^T & 0 & F_1
\end{bmatrix}
\]

4. Examples

We claim that the gluing algorithm is useful for \( s = 3 \). In fact, the \((2, t, n)\) -graphs are, \( K_n^c, n < t \). Since \( G_1 \) does not have any cliques, the condition 3 must be checked for only \( 1 \)-cliques of \( G_1 \). Therefore, the neighbors of vertices of \( G_1 \) are independent sets of \( H_1 \).

**Example 4-1** Suppose \((s, t, n) = (3, 10, 40)\). If \( G \) is a \((3, 10, 40)\) -graph and \( v \in V(G) \), then \( G_1 \) is a \((2, 10, d)\) -graph and \( H_1 \) is a \((3, 9, 39 - d)\) -graph. Thus \( 3 < d < 10 \).

Let \( d = 4 \) and \( H_1 \) be a \((3, 9, 35)\) -graph. The vertex set of \( G_1 \) is the set \( \{v_1, v_2, v_3, v_4\} \).

Let \( N_i \) be the neighbors of \( v_i \) in \( H_1 \), \( i = 1, 2, 3, 4 \). Thus we have:

**Condition 3.** The \( N_i \)s are independent sets for \( i = 1, 2, 3, 4 \).

**Condition 4.** Let \( l = 1, 2, 3, 4 \). We have the following conditions:

- **Case 1.** For \( i = 1, 2, 3, 4 \), \( H \setminus N_i \) does not have any independent set of order 9. This case holds trivially.
- **Case 2.** For \( i, j = 1, 2, 3, 4, i \neq j \), \( H \setminus (N_i \cup N_j) \) does not have any independent set of order 8.
Case3. For different \( i, j, k = 1, 2, 3, 4 \), \( H \setminus \left( N_i \cup N_j \cup N_k \right) \) does not have any independent set of order 7.

Case4. \( H \setminus \left( N_1 \cup N_2 \cup N_3 \cup N_4 \right) \) does not have any independent set of order 7.

In the greedy algorithm, the Kid graph has similar structure like Fa. The following example use this algorithm for generating the \((5, 5, 43)\)-graphs.

Example 4-2 Suppose \((s, t, n) = (5, 5, 43)\). Let \( Fa, Mo \) be the two graphs with 43 vertices. In fact, we use the adjacency matrices of these graphs. Then we enumerate all subsets of order 5 from \( \{1, 2, \cdots, 43\} \).

Now, we choose all subgraphs of order 5 from \( Fa, Mo \). These subgraphs are induced \((0, 1)\)-submatrices of order 5. These subgraphs are 5-cliques if and only if the sum of their entries are 10 and they are independent sets of order 5 if and only if the sum of their entries are 0. Thus we enumerate the 5-cliques and independent sets of order 5 from \( Fa, Mo \).

We use the \texttt{FindMaxim(Fa)}, \texttt{FindMaxim(Mo)} to find the maximum subgraphs (or submatrices) of \( Fa, Mo \) without 5-cliques and independent set of order 5. We called these graphs are \( F, M \). If \( \vartheta (F) + \vartheta (M) < 43 \), choose some vertices from \( Fa \) to find a \( 43 \times 43 \)-matrix. This is the \texttt{KidMake(F,M)} procedure. Now we enumerate the 5-cliques and independent sets of order 5 from \texttt{Kid}. Thus \texttt{KidMake(F,M)} must be construct the new graphs with the number of 5-cliques and independent set of order 5 are less than their parents.

5. Conclusion

The gluing algorithm find \((s, t, n)\)-graph if and only if you have a complete database of \((s - 1, t)\)-graphs and \((s, t - 1)\)-graphs.

Let \( s = 3 \). Therefore, \((2, t)\)-graph database are the known graphs: independent sets.

Thus the gluing algorithm of these graphs to \((3, 9)\)-graphs is easy to use.

In the greedy algorithm, it is possible to find Kid regular graph if \( Fa, Mo \) are regular.

This algorithm is so fast, but its solutions are not \((s, t, n)\)-graphs, but approximately \((s, t, n)\)-graphs. One could hope in several generation, the Kid will be a \((s, t, n)\)-graph!

Since the most famous \((s, t, n)\)-graphs are regular, these algorithms could be refined to construct regular graphs from regular parents. The critical \((s, s, n)\)-graphs are self-complementary graphs. One could use some versions of the greedy algorithm to construct self-complementary graphs.

References


http://dx.doi.org/10.1002/jgt.3190190304.