# Synthesis of Arithmetical Programs for Ramsey Graphs

Thibault Gauthier

Czech Technical University in Prague, Czech Republic

**Abstract**

We present an approach for constructing Ramsey graphs based on the synthesis of arithmetical programs. Our algorithm has created a simple formula that is conjectured to produce an exponential lower bound for diagonal Ramsey numbers.

**Introduction** It is possible to automatically create formulas to solve a open problem in mathematics. For example in 1995, Simon Plouffe discovered with the help of a computer a formula for generating the $n^{th}$ digits of $\pi$ efficiently [1]. In this talk, we investigate the problem of finding an exponential asymptotic lower bound for diagonal Ramsey numbers. In 1947, using a probabilistic method, Erdös proved [3] an asymptotic lower bound for the diagonal Ramsey numbers $R(s,s)$ which implies that for $s > 5$, $R(s,s) > (\sqrt{2})^s$. Yet, there are no known explicit constructions from which one could prove an exponential lower bound (i.e. $R(s,s) > (1+\epsilon)^s$). Therefore, we present a method for creating candidate constructions from arithmetical programs. To do so, we adapt our system for synthesizing programs from integer sequences [5] which so far has discovered programs for more than 120,000 OEIS sequences [6]. We also draw some of our intuitions about the nature of this problem from our recent formalization [4] of $R(4,5) = 25$.

**An Arithmetical Programming Language** The programs are synthesized using a programming language containing a few primitives. This facilitates the learning and encourages the system to come up with its own solutions. It contains the constants 0, 1, 2, the functions $+$, $-$, $\times$, *div*, *mod*, two arbitrary-precision integers variables $i, j$, the conditional operator *cond*, three looping operators *loop*, *loop2*, *compr* and two list operators *push*, *pop*.

**From Arithmetical Programs to Ramsey Graphs** To generate a labeled graph $G_n$ of size $n$, we take a program p that represents a function $f : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ and compute $f(i,j)$ for all $i$ and $j$ such that $0 \leq i < j < n$. There is an edge in $G$ between a vertex $i$ and a vertex $j$ such that $i < j$ if and only if $f(i,j) > 0$. Note that by increasing $n$ the program $f$ can be used to construct a series $(G_n)_{n\in\mathbb{N}}$. We say that an element of the series $G_n$ respect the asymptotic constraint if it has a maximum clique of size $s$ and a maximum independent set of size $t$ and $n > (\sqrt{2})^{max(s,t)}$.

**Our Approach** To find programs generating Ramsey graphs, our approach relies on a self-learning loop that alternates between three phases. During the search phase, a neural network synthesizes millions of candidate programs. If two programs generate the same labeled graph, only the smallest program is kept as a candidate. Then, during the checking phase, a scoring function is used to rank the candidate programs. The score of a series $(G_n)_{n\in\mathbb{N}}$ is the first size $n \leq 64$ for which $G_n$ fails to respect the constraint. We give higher scores to shorter programs to break ties. In the learning phase, a neural network is trained to reproduce the 10,000 best programs discovered so far. Each iteration of the self-learning loop leads to the discovery of programs with better scores.

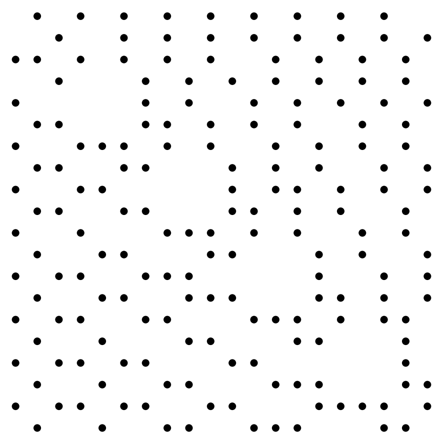

Figure 1: Adjacency matrix of a generated graph

**Results** After a few generations, all 10,000 best programs respect the asymptotic constraint up until size 64. Further checking shows that more than 1,000 programs respect the constraint up until size 256. Among those, we searched for programs that did not contain any loops or any mention of integer division as we believe it would be easier to prove the asymptotic behavior of such programs. We found only one such program. Given $i < j$, there is an edge between $i$ and $j$ if and only if:

$$(j^2 \mod (2(i+j)+1)) - (i+j) > 0$$

The adjacency matrix of this graph on its first 20 vertices is given in Figure 1 revealing an organized structure. After further checking, we confirmed that this program produces a $\mathcal{R}(20, 20)$-graph of size 1024. To efficiently check for the absence of cliques (independent sets) in such a large graph, we adapt a branch and bound algorithm for finding a maximum clique in a graph [2].

**Future Works** According to this first experiment, it seems relatively easy to synthesize formulas that construct Ramsey graphs with an observed exponential asymptotic lower bound. However, proving such observation is much harder. To simplify the potential proof, we are considering two orthogonal ideas. The first one is to force the constructions to respect extra constraints such as graph symmetries or regularities in the distributions of cliques. The second one is to produce family of graphs using an alternative construction. Starting from a graph of size $2^n$, one can construct of size $2^{n+1}$ by duplicating the graph and adding transverse edges according to the return value of a synthesized program.

**Resources** The code for our project is available in this repository `https://github.com/barakeel/oeis-synthesis`. The most relevant part is located in the file `ramsey.sml`.

# References

[1] David Bailey, Peter B. Borwein, and Simon Plouffe. On the rapid computation of various polylogarithmic constants. *Math. Comput.*, 66(218):903–913, 1997.

[2] Randy Carraghan and Panos M Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9(6):375–382, 1990.

[3] Paul Erdös. Some remarks on the theory of graphs. 1947.

[4] Thibault Gauthier and Chad E. Brown. A formal proof of R(4, 5)=25. *CoRR*, abs/2404.01761, 2024.

[5] Thibault Gauthier, Miroslav Olsák, and Josef Urban. Alien coding. *Int. J. Approx. Reason.*, 162:109009, 2023.

[6] Neil J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. In Manuel Kauers, Manfred Kerber, Robert Miner, and Wolfgang Windsteiger, editors, *Towards Mechanized Mathematical Assistants, 14th Symposium, Calculemus 2007, 6th International Conference, MKM 2007, Hagenberg, Austria, June 27-30, 2007, Proceedings*, volume 4573 of *Lecture Notes in Computer Science*, page 130. Springer, 2007.