AI-powered Discovery Of Counterexamples in Combinatorics Final Report

2025-07-30

Student Researchers: Robert Sun | Thomas Kidane | Kate Newbold | Iurii Beliaev | Yasir Alhasaniyyah | Kevin Wang
Project Leader: Fan Wei
Project Manager: Raymond Sun

Abstract

Our research leverages cutting-edge machine learning and computational search techniques to tackle longstanding open problems in mathematics. By reframing abstract conjectures as optimization or search tasks, we use AI to explore vast possibility spaces and hunt for elusive counterexamples. Through the development of an efficient search framework, we contributed new approaches and evidence relevant to the Sidorenko Conjecture, the Second Neighborhood Conjecture and the universal existence of envy-free up to any item (EFX) allocations. Our methods are broadly applicable and hold promise in tackling other challenging problems in combinatorics.

Background

1. Sidorenko's Conjecture [1] is a major conjecture in the field of extremal graph theory, it was posed by Alexander Sidorenko in 1986 and its general case remains open to this day.

   Formally, it states that every bipartite graph H has Sidorenko's property, i.e. for all graphons W, the inequality $t(H,W) \geq t(K_2,W)^{|E(H)|}$ is true, where $t(H,W) = \int_{[0,1]^{|V(H)|}} \prod_{ij \in E(H)} W(x_i,x_j) \prod_{i \in V(H)} dx_i$ is the homomorphism density. An equivalent reformulation which uses finite graphs instead of graphons is that if a graph G has n vertices and an average degree of pn, then $t(H,G) \geq p^{|E(H)|}$.

   An interpretation of this inequality is that the density of copies of H in G is asymptotically minimized by a random graph, since if each edge exists with probability p, the fraction of possible subgraph copies of H would be $p^{|E(H)|}$.

   Sidorenko's conjecture has been proved for many special cases such as paths, trees, and cycles of even length. The smallest graph H for which Sidorenko's conjecture is still open is the Mobius Ladder $K_{5,5} \setminus C_{10}$, which is the main focus of our search for counterexamples since it can be computed most efficiently.

2. The Second Neighborhood Conjecture [2] is another long-standing conjecture in graph theory, posed by Paul Seymour in 1990. In an oriented graph, the first neighborhood of a vertex v is defined as $N^{+}(v)$, the set of vertices at distance 1 from v. The second neighborhood of v is then defined as $N^{++}(v)$, the set of vertices at distance 2 from v, which is disjoint with the first neighborhood. A vertex v is called a Seymour vertex if its second neighborhood is at least as large as its first neighborhood. The Second Neighborhood Conjecture states that in every oriented graph with no two-edge cycles, there always exists at least one Seymour vertex.
3. The problem of fair allocation of indivisible items can be described with a utility function $v_i(g)$for each agent i and item g, which gives a valuation of how strongly they want g. Once items are allocated, an agent envies another agent's bundle if the value they give that bundle is higher than the value they give their own. Although it's easy to show that envy-free allocations do not exist in general, there are many relaxations that have been proposed, with EFX being a significant one of them. An allocation is called EFX if for every two agents A and B, if we remove *any* item from the bundle of B, then A does not envy B . A well-known open question is whether the EFX allocation exists in general, and we try to either find a counterexample, or propose an algorithm that can always construct an EFX allocation. [3]

These three open questions were chosen because they are relatively well-known in their respective fields, they have resisted traditional mathematical techniques for decades, and their potential counterexamples can be easily verified by computation. Despite their prominence, there have been few documented efforts to explore these conjectures using AI-powered methods. Through our work, we aim to uncover underlying structures that may contribute to a deeper understanding of these longstanding challenges.

Methods & Results

1. **Reward Functions**

   The main idea we use to search for counterexamples comes from an approach proposed by Wagner in 2021 to disprove conjectures in graph theory [4], which frames a conjecture as a reward function $f(G) < 0$ and uses reinforcement learning (RL) to maximize the final score $f(G_T)$ through playing a graph-building game. The game is played by adding or removing edges to a graph in each step, and the priority is to find the graph with a positive score which would be a counterexample. In the case of Sidorenko's conjecture, two natural choices for the reward function would be $p^{|E(H)|} - t(H,G)$ (Sidorenko Gap) and

   $\frac{t(H,G)}{p^{|E(H)|}} - 1$ (Sidorenko Ratio). Since the Sidorenko Gap does not scale with p, using this reward function causes learning to become biased towards sparser and sparser graphs. We also tested some other reward functions, but they did not perform better than the more intuitive Sidorenko Ratio. Therefore, the reward function used below for Sidorneko's Conjecture is $f_R(H,G) = \frac{t(H,G)}{p^{|E(H)|}} - 1$.

   For the second neighborhood conjecture, $min_v|N^{+}(v) - N^{++}(v)|$ is a natural reward function. However, this would easily be dominated by a single Seymour vertex where the second neighborhood is large, and it would be very inefficient at removing other Seymour vertices. Therefore, the reward function we use is $-N_S$, where N_S is the amount of Seymour vertices in the graph. Only using this reward could still cause the graph building game to easily get stuck in local maximums, so we added other penalties based on known findings.

   For the EFX conjecture, a potential reward function is $\min_X \{\max_{i,j,g} [v_i(X_j \setminus g) - v_i(X_i)]\}$. However, this takes computations on the order of m^n to compute the reward function for m agents and n items, which makes it too computationally expensive to perform searches on for cases larger than 5 agents and 10 items. In addition, this reward function reaches 0 when the utility functions for all agents are identical, but this is a case where EFX is proven to exist and isn't actually close to a counterexample. Therefore, it became clear that directly searching for counterexamples isn't practical for EFX, and we shifted our focus towards finding an algorithm to prove EFX exists instead.

2. **Sidorenko's Conjecture**

We began our search for counterexamples of Sidorenko's conjecture using its equivalent reformation, and the most glaring barrier was computing $t(H,G)$. We found an online package that could count homomorphism densities using dynamic programming, which was over 100 times faster than brute force counting. However, its memory limit caused inaccurate counts for G with over 8 vertices, so we fixed the code and implemented an accurate C++ library. With this library, we were able to use RL to search for the best G for graphs up to 15 vertices. In numerous attempts, the graph G would always converge to the complete graph K_n, and no violation to the conjecture was found.

Since it is possible that a counterexample would lie in cases with G having more than 15 vertices, we then decided to focus on the original graphon version of Sidorenko's Conjecture, since graphons can approximate infinite graphs with a certain structure, with weights of a nxn matrix representing probability of edges existing between groups of vertices. To compute homomorphism density for graphons $t(H,W)$, we also updated our custom homomorphism counting library so that it can run in $\mathcal{O}(|V(H)| \cdot |V(G)|^{\mathrm{tw}(H)+1})$ complexity. Graphons do pose another challenge, because there is no straightforward way to modify the weight matrix the way we can add or remove edges of a finite graph. Therefore, we used the idea of gradient-based optimization, perturbing the weight matrix of the graphon and accepting the perturbation if it improves the score.

To escape local optima when progress stalls, we use Adaptive Monte Carlo Search (AMCS). [5] This combines a fast local search with a strategic outer loop that increases search depth when it gets stuck. To enhance search speed and prevent the final matrix to depend on initial weights, we start by taking large perturbation steps with L_2 norm of 0.3. We then decrease the step length ϵ when no improvement is made and $\epsilon > \epsilon_{min}$, then increase it once an improvement is found. The search is ended when either no improvement has been made for 3 consecutive loops, or if the weight matrix is too close to uniform, since it has been proven that Sidorenko's Conjecture holds for a graphon W where $\|W - 1\|_\infty \leq \frac{1}{60}$. [6] To make the Sidorenko Ratio comparable between steps, we normalize the weight matrix so that the weights average to 1.

Using AMCS for the 4*4 graphon case, the weight matrix would eventually converge towards uniform weights, with the rate of convergence depending on the parameters of AMCS, usually stablizing in around
15 minutes and exiting the loop in around an hour. The following is an example of the printed results of AMCS, where W_{opt} is the best matrix found:

$$W_{\text{initial}} = \begin{bmatrix} 0.1 & 1.1 & 1.3 & 0.8 \\ 1.1 & 0.5 & 0.4 & 1.1 \\ 1.3 & 0.4 & 0.9 & 1.2 \\ 0.8 & 1.1 & 1.2 & 1.4 \end{bmatrix},$$

Best Sidorenko Ratio: -2.9797×10^{-8}

Total time: 3336.68 seconds

$\epsilon_{min} = 0.0005$

$$W_{opt} = \begin{bmatrix} 0.9871 & 1.0128 & 0.9895 & 1.0106 \\ 1.0128 & 1.0103 & 0.9933 & 0.9835 \\ 0.9895 & 0.9933 & 1.0167 & 1.0004 \\ 1.0106 & 0.9835 & 1.0004 & 1.0055 \end{bmatrix}$$

While our searches produced W_{opt} with a Sidorenko Ratio on the order of -10^{-8}, they still didn't find counterexamples, which could be due to the limitations of a 4*4 graphon structure. Using this method for a larger graphon such as 8*8 would take too long, so we used a spectral version that could be applied specifically for $H = K_{5,5} \setminus C_{10}$ for these larger graphons. [7] Through our optimization attempts, we noticed that while the weights always eventually converge to 1,

the row sums and column sums of the normalized W_{opt} matrix converges to 4 much quicker than the individual weight entries converge to 1, so these matrices may contain structures that make them closer to counterexamples.

We also attempted reinforcement learning strategies such as proximal policy optimization, but one obstacle is that it becomes difficult to learn a meaningful policy in a 64-dimension continuous space of weights. Therefore, we use a hybrid AMCS+RL method which does global exploration to identify a promising region of the solution space with RL, then utilizes AMCS locates a superior local optimum. While no counterexample was found with this method, it did find a matrix that came closer to violating the conjecture than RL or AMCS alone.

Method	**Conjecture Gap**	**Time (s)**
RL Alone	5.70×10^{-8}	143.18
AMCS Alone	8.23×10^{-11}	328.58
Hybrid	$\mathbf{6.08 \times 10^{-15}}$	**153.08**

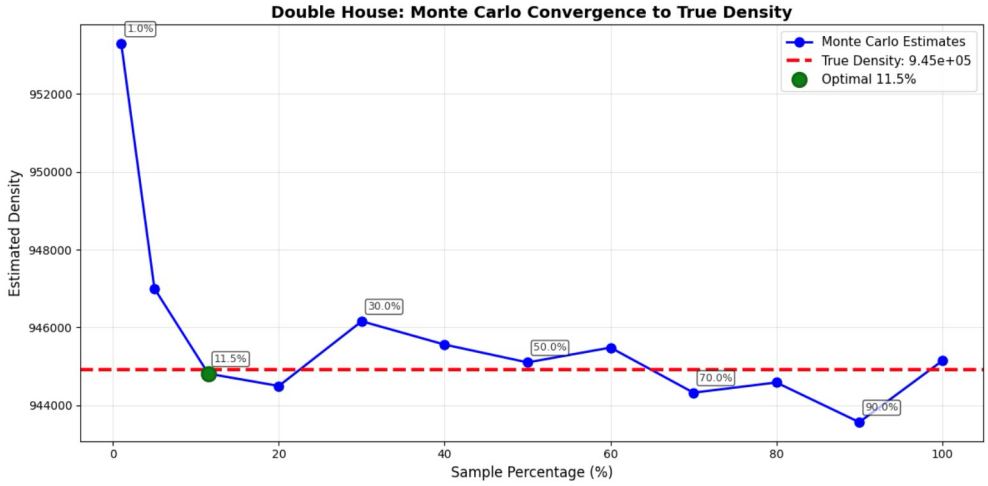
Table 1: Hybrid search finds superior solutions

3. **Approximations**

Another approach we attempted to increase search efficiency was approximating homomorphism densities instead of computing the exact value. Before applying it on searches, we first compared our approximation results with true results for smaller graphs to determine what percentage to sample.

We trained a graph attention network (GAT) to model homomorphism density distribution for a fixed target graph, which predicts homomorphism density with 1.06% maximum and 0.21% average error. This shows promise, and we can try to incorporate more sophisticated Graph Neural Networks (GCN, GAT) to improve our model's expressivity and use transfer learning to scale our search to larger, more complex graphs.

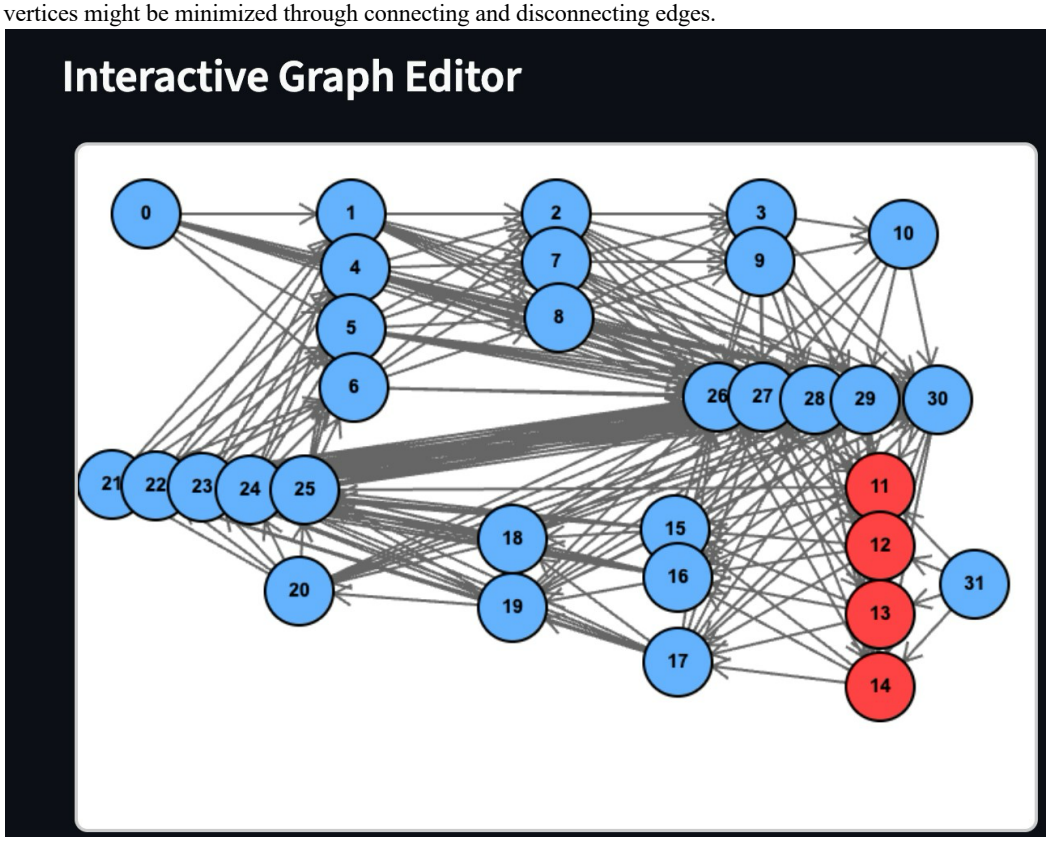
Another way of approximation we tried for large graphs was Monte Carlo sampling. Through testing, we found that sampling around 11.5% of total homomorphisms would achieve the best balance between computational efficiency and estimation accuracy for most graphs.



4. **Second Neighborhood Conjecture:**

   Our initial searches tended to evolve graphs toward structures with a single "sink" vertex, which trivially satisfies the conjecture and suggests it is quite robust. Since this conjecture has been proven to be true for any oriented graph with at least one vertex of out-degree no greater than 6, we also must punish graphs for having vertices with low out-degree. However, in our attempts, a large penalty for low out-degree vertices would generally cause the final amount of Seymour vertices to be large, sometimes even reaching 50% of vertices in the graph, which is not ideal for finding a counterexample. Therefore, we used evolutionary search with a more balanced penalty that can both reduce the amount of Seymour vertices and prevent the graph from falling into cases where the conjecture is already proved.

   After using evolutionary searches, our best-performing graphs tended to be dense in the core, sparse in the periphery and contain at least one vertex with a very large second neighborhood. In addition to RL, we also designed an interactive graph editor app: https://mathresearch.streamlit.app/, where we can build a custom graph ourselves. Seymour vertices are colored red while other vertices are colored blue, so we can visually understand how the number of Seymour vertices might be minimized through connecting and disconnecting edges.



5. **EFX and MM Allocations in Fair Division**:

   In fair allocation, there always exists an allocation called the Maximin (MM) or egalitarian allocation, defined as the allocation in which the smallest total utility of an agent is as large as possible. [8] Since most MM allocations are also EFX allocations, [9] we intend to find an algorithm to produce EFX allocations universally starting from a MM allocation. We start from using programs to search for counterexamples, specifically ones that are MM but not EFX. Since there have been results proving that EFX always exists when agents only have 3 types of utility functions, [10]

we also look specifically at cases where agents have the same ranking preference of items (differing only in valuation magnitude) as a natural extension.

We employ a three-stage evolutionary search to search for counterexamples:

1. Generate a population of random valuation matrices
2. Evaluate the MM allocation for each matrix and check if it is EFX.
3. Evolve the fittest matrices, defined by number of close EFX allocations, by mutation through slightly altering each agent valuation.

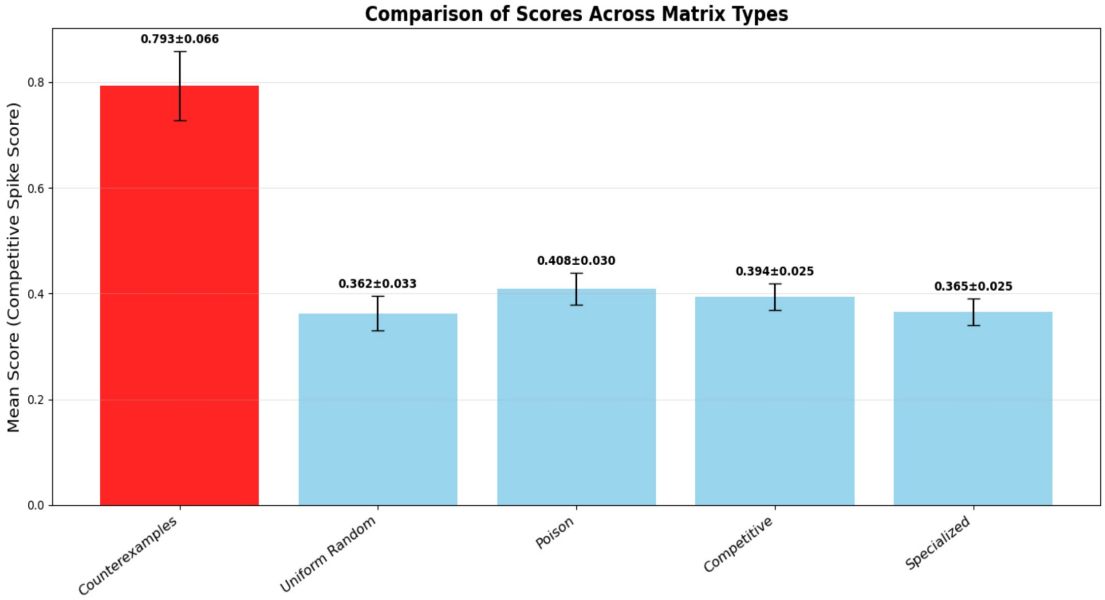
This algorithm yielded several key findings:

The evolutionary search successfully and consistently generated counterexamples where none of its MM allocations are EFX. It also uncovered counterexamples in utility matrices where all agents shared the same item ranking.

Many of the counterexamples exhibited clear structural patterns. One common feature was the presence of *competitive goods*—one or two items that were valued exceptionally highly by multiple agents. Another recurring pattern was *spiky preferences*: although less pronounced in the second wave of counterexamples with identical item rankings, these cases often included at least one agent with strong preferences for specific items, making compromise significantly more difficult. This level is quantified by the H-score:

$$\text{H-score}(\mathbf{v}) = \frac{1}{2n\sum_{i=1}^{n} v_i} \sum_{i=1}^{n} \sum_{nj=1} |v_i - v_j|$$

The following bar graph shows how counterexamples have significantly higher H-scores than other utility matrices with a high confidence level.



The following is an example of one counterexample found by the evolutionary search, where there are 3 agents and 5 items to allocate.

	Item 1	**Item 2**	**Item 3**	**Item 4**	**Item 5**
Agent a	94	18	12	13	2

Agent b	95	11	12	14	13
Agent c	7	9	19	3	18

Table 2: Example of a Counterexample Valuation Matrix

Only Relative MM Allocation

- Agent a: {2, 3, 4} (30.94% total utility)
- Agent b: {1} (65.52% total utility)
- Agent c: {5} (32.14% total utility)

EFX Allocation (Optimized for Max-Min)

- Agent a: {2, 4} (22.30% total utility)
- Agent b: {1} (65.52% total utility)
- Agent c: {3, 5} (66.07% total utility)

It can be seen that while the only MM allocation for this example is not an EFX allocation, it is just one swap away from the closest EFX allocation, by agent a giving item 3 to agent c . Another example where the MM allocation is not EFX even when all agents shared the same item ranking is shown below:

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
a	527	571	3876	4300	4716	6106	6840	7105	8510
b	1741	3687	3762	6120	6490	8054	8251	8901	9953
c	48	156	355	853	1084	4606	5834	8997	9574
d	638	900	1360	2798	3053	3449	5694	6962	8376

Table 3: Other Example of Counterexample Valuation Matrix (Every Agent has the Same Ranking of Items)

Only Relative MM Allocation:

- Agent a: $\{i_1, i_2\}$ (30.42% of total utility)
- Agent b: $\{i_3, i_4, i_5\}$ (28.61% of total utility)
- Agent c: $\{i_6, i_7\}$ (28.71% of total utility)
- Agent d: $\{i_8, i_9\}$ (29.30% of total utility)

Best EFX Allocation (Optimized for Max-min):

- Agent a: $\{i_1, i_2\}$ (30.42% of total utility)
- Agent b: $\{i_6, i_3, i_8, i_5\}$ (27.53% of total utility)
- Agent c: $\{i_9\}$ (30.39% of total utility)
- Agent d: $\{i_4, i_7\}$ (29.37% of total utility)

Once a counterexample is found, we can then analyze the distance to every "optimal" EFX allocation, as well as sub-optimal EFX allocations. The following distance metrics could used for analysis:

1. Swap Distance: Minimum trades from MM to EFX allocation
2. Euclidean Distance: Normalized change in utilities
3. Hamming Distances: Number of items reassigned

Conclusions

Our research demonstrates that reframing mathematical conjectures as optimization problems creates a powerful avenue for exploration using AI. We developed a novel hybrid algorithm combining Reinforcement Learning and Adaptive Monte Carlo Search, which proved its effectiveness in comparison to its predecessors on the Sidorenko conjecture. Simultaneously, we found that for the Second Neighborhood conjecture evolutionary search consistently evolves graphs towards structures that are closer to counterexamples. Tools we developed such as the homomorphism counting library for graphons and GATs for approximation could potentially be reused in future research, making computation more efficient for large graphs.

While for most matrices some MM allocation is always EFX, this is not a sufficient condition, and the two fairness criteria are not always compatible. We provide a computation framework to show this tension and where conflict arises. Evolutionary search is a highly effective method for navigating the complex landscape of fair division problems and identifying instances. The hardest problems arise from specific structures (although this is being further investigated depending on ranking of items). Our work provides empirical evidence and new insights into the boundaries of fair division theory.

Future Work

Given that all our optimization attempts for Sidorenko's Conjecture where $H = K_{5,5} \setminus C_{10}$ have converged toward a matrix with uniform weights, it may be more promising to try to prove that the conjecture is true for the class of Mobius Ladders instead. The spectral version of Sidorenko's Conjecture could help with the proof, but tighter bounds would need to be established on matrix eigenvalues for a rigorous proof to be completed.

For the Second Neighborhood Conjecture, a potential strategy going forward is to use the structured graphs with few Seymour vertices we found, and try to design a patching algorithm that can patch them into true counterexamples.

To continue our work on the relation between MM allocations and EFX allocations, we could utilize RLbased agents to act as more sophisticated "hunters" for counterexamples or to learn optimal allocation policies directly. Our current metric is effective in indicating counterexamples for "normal-looking" matrices, but edge cases can shift data and create false positives, so updated score testing could help as well. We could also design and test efficient swap-based algorithms to construct an EFX allocation from a MM allocation for these counterexamples.

In addition to searching for counterexamples, another promising application of AI tools lies in improving existing bounds for constructions in combinatorics. Two other graph theory problems we have explored include improving lower bounds on the most induced cycles of a n-vertex graph [11] and the domination exponent for pairs of graphs. [12] Our hybrid search framework is applicable to these problems, and we are also enthusiastic about leveraging cutting-edge large language models (LLMs) to tackle them. Google DeepMind's AlphaEvolve (2025) and Funsearch (2023) have already demonstrated potential in evolving current codes to construct better solutions to these problems. While reinforcement learning often struggles with sparse rewards, local optima and noisy gradient signals, FunSearch effectively evolves diverse candidates while retaining memory of best results, enabling more flexible constructions for improving lower bounds. Given the rapid advancement of large language models, it is likely they will soon play a significant role in accelerating mathematical research.

References

[1] Sidorenko, A. F. (1993). A correlation inequality for bipartite graphs. *Graphs and Combinatorics, 9*(1), 201–204. https://doi.org/10.1007/BF01202370

[2] Seymour, P. (Conjecture). See "The second neighborhood problem" by DeVos, et al.

[3] Chaudhury, B. R., Garg, J., & Mehlhorn, K. (2020). *EFX exists for three agents*. CoRR, abs/2002.05119. https://doi.org/10.48550/arXiv.2002.05119

[4] Angileri, F., Lombardi, G., Fois, A., Faraone, R., Metta, C., Salvi, M., Bianchi, L. A., Fantozzi, M., Galfrè, S. G., Pavesi, D., Parton, M., & Morandin, F. (2024). *A systematization of the Wagner framework: Graph theory conjectures and reinforcement learning* (CoRR, abs/2406.12667). arXiv. https://doi.org/10.48550/arX iv.2406.12667

[5] Vito, V., & Stefanus, L. Y. (2023). *Adaptive Monte Carlo search for conjecture refutation in graph theory*. CoRR, abs/2306.07956. https://doi.org/10.48550/arXiv.2306.07956

[6] Lovász, L. (2011). *Subgraph densities in signed graphons and the local Sidorenko conjecture*. *The Electronic Journal of Combinatorics, 18*(1), PaperP127, 21pp. Retrieved from http://eudml.org/doc/225090

[7] Rossman, B. (2025, May 8). *On Sidorenko's Conjecture for Bipartite Möbius Ladders* (preprint). Duke University. Retrieved from https://users.cs.duke.edu/~br148/sidorenko-mobius.pdf

[8] Brams, S. J., Kilgour, D. M., & Klamler, C. (2021). *Two-person fair division of indivisible items: Compatible and incompatible properties*. New York University. https://as.nyu.edu/content/dam/nyu-as/politics/documents/Compatibilities8.pdf

[9] Akrami, H., & Rathi, N. (2024). *Achieving maximin share and EFX/EF1 guarantees simultaneously*. CoRR, abs/2409.01963. https://arxiv.org/abs/2409.01963

[10] Prakash, H. V., Ghosal, P., Nimbhorkar, P., & Varma, N. (2024). *EFX exists for three types of agents*. CoRR, abs/2410.13580. https://arxiv.org/abs/2410.13580

[11] Král', D., Norin, S., & Volec, J. (2019). *A bound on the inducibility of cycles*. *Journal of Combinatorial Theory, Series A, 161*, 359–363. https://doi.org/10.1016/j.jcta.2018.08.003

[12] Blekherman, G., Raymond, A., Razborov, A., & Wei, F. (2023)**.** On domination exponents for pairs of graphs. *Journal of Graph Theory, 103*(3), 453–474. https://arxiv.org/pdf/2506.12151

Acknowledgements

We are grateful to Duke University for making this project possible through its generous support, facilities, and funding. Special thanks to the Data+ and Math+ programs for fostering a collaborative research environment, and to all mentors and friends whose guidance and insightful discussions were invaluable to us throughout the course of this project.