CS433: Automated Reasoning

Assignment 3: On Packing Chromatic Number of Infinite Graphs

Deadline: Monday, 31st March 2025, 11:59 PM

This assignment involves calculating bounds on the packing chromatic number of some infinite graphs by using a SAT solver. You will have to study the packing chromatic number problem, implement and optimize boolean encodings for the problem on the given graphs, and then use a SAT solver to verify your lower and upper bounds. Note that since the SAT solver might take a very long time to run, it is important to start the assignment early (or optimize your encoding well).

Please read this whole file before starting the assignment. The expected deliverables for each task of the assignment are mentioned in the *Submission* section. The formats for the encoding and proofs are mentioned in the *File Formats* section.

Reference Paper

A major part of this assignment is based on the following paper - The Packing Chromatic Number of the Infinite Square Grid is 15 by Subercaseaux and Heule (2023). The paper is available at https://arxiv.org/pdf/2301.09757.

Please go through the paper before starting the assignment.

Definitions

A packing k-coloring of a simple undirected graph $G = (V, E)$ is a function $f : V(G) \to \{1, 2, \ldots, k\}$ such that for any two distinct vertices u and v of G with $f(u) = f(v) = c$, the minimum distance between u and v is at least c.

The packing chromatic number of a graph G, denoted by $\chi_p(G)$, is the minimum number k such that G has a packing k-coloring.

Bounds on the packing chromatic number of a graph can be verified as follows: - To verify a lower bound (i.e. to show that $\chi_p(G) > k$), we need to show that there is no packing k-coloring of G. This can be done by encoding the existence of a packing k-coloring of G as a SAT problem and then providing a proof of unsatisfiability. - To verify an upper bound (i.e. to show that $\chi_p(G) \leq k$), we need to provide a packing k-coloring of G. This coloring can be found using a SAT solver. For infinite graphs, we can provide a coloring for a finite subgraph of G and then replicate this coloring infinitely.

Problem Statement

The assignment is divided into 3 tasks. The first task involves the linear infinite graph ($\mathbb{Z}^1$), the second task involves the infinite square grid graph ($\mathbb{Z}^2$), and the third task involves the infinite cubic grid graph ($\mathbb{Z}^3$).

Task 0: Linear Infinite Graph ($\mathbb{Z}^1$)

This is a warm-up task to get you started with the problem.

1. Convince yourself that the packing chromatic number of the linear infinite graph is 3.
2. Implement a SAT encoding for the coloring constraints of the graph.
3. For the lower bound, show that the graph cannot be colored with 2 colors.
4. For the upper bound, provide a coloring of the graph with 3 colors using a SAT solver or any other method.

In your report, provide and explain the encoding you used for ($k = 2$) as well as the coloring you found for ($k = 3$).

Task 1: Infinite Square Grid Graph ($\mathbb{Z}^2$)

There are 2 subtasks in this task. For this task, more marks will be awarded for tighter bounds.

Subtask 1.1: (Highest) Lower Bound

1. Implement a SAT encoding for the coloring constraints of the infinite square grid graph with some $k < 15$ colors.
2. Generate a proof of unsatisfiability to show that the graph cannot be colored with k colors.

In your report, provide and explain the encoding you used for the lower bound. Explain in short any optimizations you made during the encoding process.

Subtask 1.2: (Lowest) Upper Bound Obtain a coloring of the infinite square grid graph with some $k \geq 15$ colors.

In your report, provide the coloring you found for the upper bound.

Task 2: Infinite Cubic Grid Graph ($\mathbb{Z}^3$)

This is the main task for this assignment. Note that it is proven that the graph does not have a packing k-coloring for any finite k. We can hence generate a proof of unsatisfiability for any k.

Subtask 2.1: Try an Optimization

1. Read section 3 of the reference paper to understand some possible optimizations for boolean encoding of the infinite square grid graph.

2. Select any 1 of those optimizations and apply/extend it to the infinite cubic grid graph.
3. Select atleast 10 different values of k and illustrate the impact of the selected optimization on the time taken by the SAT solver to return an `UNSAT` result.

In your report, provide and explain the encoding you used. Mention the optimization you selected and explain how it works in both graphs. Provide plots to show the speedup achieved by the optimization. Comment on the applicability of the optimization on the infinite cubic grid graph.

Subtask 2.2: Create an Optimization

1. Propose atleast 1 new optimization for the boolean encoding of the infinite cubic grid graph.
2. Select atleast 10 different values of k and illustrate the speedup achieved by the proposed optimization.

In your report, provide and explain the encoding you used with all the proposed optimizations. Provide plots to show the speedup achieved by each specific optimization. Comment on the effectiveness of the proposed optimizations.

Subtask 2.3: (Highest) Lower Bound Again

1. Using the optimizations from the earlier subtasks, implement a SAT encoding for the coloring constraints of the infinite cubic grid graph with some k colors.
2. Generate a proof of unsatisfiability to show that the graph cannot be colored with k colors.

Select the highest value of k for which you can generate a proof of unsatisfiability using reasonable computational resources. In your report, provide and explain the encoding you used for the lower bound. Explain in short any optimizations you made.

Submission

The submission should include one file named `report.pdf` which should contain the explanations, results, and plots for all the tasks. Any citations should be explicitly mentioned in the report. Note that the Juptyer notebooks are only needed for generating the encodings. For the actual SAT solving, you can use any SAT solver of your choice as well as its bindings in any programming language.

Other files that should be as follows: - Task 0: - Encoding for $k = 2$ in a file named `task0_enc.cnf`. - Proof of unsatisfiability for $k = 2$ in a file named `task0_proof.txt`. - Juptyer notebook containing the code to generate the encoding. - Task 1: - Encoding for the lower bound in a file named `task1_enc_k.cnf` where `k` is the value of k you used. - Proof of unsatisfiability for the lower bound in a file named task1_proof_k.txt. - Juptyer notebook containing the code to generate the encoding. - Task 2: - All the encodings used in report for both unoptimized and optimized versions in files named task2_enc_k_*.cnf where k is the number of colors in this encoding. - Proof of unsatisfiability for the corresponding encodings in files named task2_proof_k_*.txt. - Juptyer notebook containing the code to generate the encoding including the optimizations. - Please name the files aptly using the naming convention mentioned above and have same values of * in corresponding files.

Submit all these files in a compressed folder. The final tar file should be named <roll_number_0>_a3.tar.gz or <roll_number_0>_<roll_number_1>_a3.tar.gz or <roll_number_0>_<roll_number_1>_<roll_number_2>_a3.tar.gz depending on the number of members in your group. Only 1 group member should submit the file. The submission method will be shared later.

File Formats

To standardize the submission, please prefer the following formats for the files: - All encoding files (.cnf) should be in the DIMACS CNF format. - All proof files (.txt) should be in the DRAT format which can be verified with the DRAT-trim checker.

Compress all the files into a single tar file whose size does not exceed 1GB.

We are open to accepting submissions (especially for the proofs) in other formats as well. **Please write on MS Teams at least 5 days before the submission deadline if you plan to submit in a different format**. Ensure that the format is easily verifiable (via an open source tool). Using a different format without prior intimation will result in a penalty.

Citations

It is expected that you will use the reference paper as a major source for this assignment. You are free to use any other sources as well.

Ensure that you cite any sources you use in your report. In particular, if you use encodings, colorings or optimizations from any other source, you should explicitly mention the source.

Note on computational resources

It is expected that generating the proofs for the lower bounds will require significant time and computational resources. You can use multiple cores, parallelization, and multiple machines to generate the proofs. Ensure that you stich the proofs correctly and provide the complete proof in the submission.

Note that optimizing the encodings is a crucial part of this assignment. In general, it will be much better to spend time optimizing the encoding than to spend time generating the proof.