An Integer Programming Approach to Solving Rummikub Configurations

Your Name
Department of Computer Science
Your University

April 30, 2025

Abstract

Rummikub is a popular tile-based game that involves both combinatorial optimization and strategic decision-making. This paper presents a formal investigation into the game mechanics, identifies the core computational challenges, and introduces an integer programming (IP) model designed to find optimal moves for a given player state. We analyze the problem space, define constraints based on legal game rules, and propose an IP formulation to maximize score per turn. Computational experiments validate the model's effectiveness and provide insights into its potential for AI agent development.

1 Introduction

Rummikub is a classic tile-laying game blending elements of rummy and mahjong. Its complexity arises from its dynamic board, flexible tile arrangements, and constraint-based rules. While humans rely on heuristics, we aim to explore a formal mathematical model that can generate optimal plays automatically.

1.1 Motivation

Understanding the strategic depth of Rummikub can aid in creating intelligent agents for game-playing and provide case studies in integer programming applications.

2 Background and Related Work

2.1 Rummikub Game Rules Overview

Each player starts with 14 tiles. The goal is to create valid sets (groups or runs) to reduce one's hand to zero. A group is three or four tiles of the same number in different colors. A run is three or more consecutive numbers of the same color.

2.2 Computational Complexity

Previous research suggests that determining a valid move is NP-hard due to the permutations and combinations of tile arrangements. de Jong and Uiterwijk [2014].

2.3 Related AI Models

Discuss existing models or papers on game AI, such as those used in Sudoku, Scrabble, or other combinatorial games.

3 Integer Programming Model

3.1 Problem Definition

Given a hand of tiles and the current board state, determine the optimal move (or series of moves) that maximizes score.

3.2 Model Assumptions

- All tiles and their colors are encoded numerically.
- The board is flattened to a list of valid sets.
- Initial meld rules and joker rules are incorporated.

3.3 Variables and Parameters

Let x_{ijk} be a binary variable indicating if tile i is used in group/run j in position k.

Define:

- T: set of all tiles
- G: set of possible groups/runs
- B: board state
- H: hand tiles

3.4 Constraints

- Each tile used only once per turn
- Groups/runs must follow valid Rummikub rules
- At least 30 points for the initial meld (if first move)

3.5 Objective Function

$$\max \sum_{i \in H} \sum_{j \in G} \sum_{k} v_i \cdot x_{ijk}$$

where v_i is the value of tile i (equal to its number).

4 Implementation and Experiments

4.1 Tools Used

Model implemented using Python and solved with Google OR-Tools / Gurobi / CPLEX.

4.2 Test Cases

Present example game states and corresponding optimal plays derived from the model.

Figure 1: Example board configuration solved by the model.

4.3 Results

Show performance in terms of solution time, move quality, and comparison with heuristic approaches.

5 Discussion

5.1 Limitations

- Model scalability as board complexity increases
- Handling dynamic player interaction

5.2 Future Work

- Integration with other AI agents.
- Use of reinforcement learning or hybrid approaches.

6 Conclusion

We have proposed an integer programming model capable of identifying optimal Rummikub moves from a static game state. This framework serves as a step toward developing intelligent agents capable of playing competitively.

References

E. de Jong and J. Uiterwijk. Rummikub is np-complete. *ICGA Journal*, 37(3):151–157, 2014.