TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science

Multicut Game: Alliance Divider

Sheng, Yichao

Research Report

TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science

Multicut Game: Alliance Divider

Sheng, Yichao

Matriculation number: 5171752
Matriculation year: 2023

Research Report

Research Project

Supervisor

Jannik Irmai

Supervising professor

Prof. Dr. Bjoern Andres

Submitted on: 12th August 2025

TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science

Task for the preparation of a Research Report

Name:	Sheng, Yichao
Matriculation number:	5171752
Matriculation year:	2023
Title:	Multicut Game: Alliance Divider

Objectives of work

The Multicut Game project aims to transform the complex combinatorial optimization problem of graph partitioning into an engaging educational game experience. The project focuses on developing an interactive game called "Alliance Divider" that teaches players about the Multicut Problem while providing an optimal solution using Integer Linear Programming (ILP) with Gurobi solver.

The main objectives include creating an intuitive game interface that represents city-states as nodes and their relationships as weighted edges, implementing the core gameplay mechanics of cutting hostile ties while preserving friendly alliances, and developing a robust technical architecture that bridges Unity game engine with Python optimization algorithms.

Focus of work

- Research and analysis of the Multicut Problem
- Design and implementation of game mechanics
- Development of Unity-Python integration architecture
- Implementation of ILP solver with Gurobi
- User interface design and gamification elements
- Testing and optimization of the complete system

Supervisor:	Jannik Irmai
Issued on:	7th April 2025
Due date for submission:	12th August 2025

Prof. Dr. Bjoern Andres
Supervising professor

Abstract

This thesis presents the development of ”Alliance Divider”, an innovative serious game that transforms the complex Multicut Problem from combinatorial optimization into an interactive gaming experience. The game allows players to learn fundamental principles of graph theory and optimization by cutting hostile connections and preserving friendly alliances between city-states. The project demonstrates successful gamification of academic concepts through Unity-Python integration and Integer Linear Programming optimization.

Statement of authorship

I hereby certify that I have authored this document entitled *Multicut Game: Alliance Divider* independently and without undue assistance from third parties. No other than the resources and references indicated in this document have been used. I have marked both literal and accordingly adopted quotations as such. There were no additional persons involved in the intellectual preparation of the present document. I am aware that violations of this declaration may lead to subsequent withdrawal of the academic degree.

Dresden, 12th August 2025

Sheng, Yichao

Restriction note

This document entitled *Multicut Game: Alliance Divider* contains confidential data, disclosed by *COMPANY*. Publications, duplications and inspections—even in part—are prohibited without explicit permission by *COMPANY*, as well as publications about the content of this document. It may only be made accessible to the supervisor at Technische Universität Dresden, the reviewers and also the members of the examination board.

Contents

List of Figures

List of Tables

1 Acronyms and Mathematical Notation

1.1 Acronyms

In the context of the Multicut Game project, several key acronyms are used throughout this document. The Integer Linear Programming (ILP) approach is central to solving the optimization problems, while Non-deterministic Polynomial time (NP)-hard complexity characterizes the computational challenge. The game's User Interface (UI) and Heads-Up Display (HUD) provide intuitive interaction, while Application Programming Interface (API) and JavaScript Object Notation (JSON) facilitate cross-language communication.

1.2 Mathematical Symbols

The mathematical foundation of the Multicut Game project is based on graph theory, where Graph G represents the game's network structure with vertex set Vertex Set V representing city-states and edge set Edge Set E representing relationships. Each edge has an associated weight Edge Weight w representing the cost of cutting that relationship. The objective is to find a cut set Cut Set C that minimizes the total cut cost Cut Cost c while satisfying the cycle inequality constraints.

2 Introduction

This chapter introduces the Multicut Game project and its objectives. The project aims to transform the complex combinatorial optimization problem of graph partitioning into an engaging educational game experience. The following chapters will provide detailed analysis of the mathematical foundations, game design, technical implementation, and evaluation of the Alliance Divider game.

3 Background and Related Work

3.1 The Multicut Problem

The multicut problem is a combinatorial optimization problem. The objective is to cut a given graph into multiple components such that the sum of the cost of the edges that are cut is minimal [Sch03].

Let $G = (V, E)$ be an undirected graph. A set of edges $M \subseteq E$ is called a multicut of G if and only if for every cycle $C = (V_C, E_C)$ in G, it holds that $|M \cap E_C| \neq 1$. That is, no cycle contains exactly one edge from M [GVY97].

The set of all multicuts of G is denoted by $\mathcal{M}(G)$. The multicut problem is the following optimization problem:

$$\min_{M \in \mathcal{M}(G)} \sum_{e \in M} c_e$$

This problem was first studied in the late 20th century and is also known as "correlation clustering" [bansal2004] and "coalition structure generation" [demaine2006]. The multicut problem is NP-hard, meaning it is unlikely that an efficient algorithm solving it exactly exists. Nevertheless, it has many interesting applications where it produces state-of-the-art results, leading to significant research efforts in understanding it [CKR00].

The problem has found applications in various domains including:

- Image Segmentation: Probabilistic image segmentation with closedness constraints [andres2012a]
- Connectomics: Globally optimal closed-surface segmentation for connectomics [andres2012b]
- Network Design: Oblivious network design and optimization [GHR04]
- Social Network Analysis: Correlation clustering and community detection [charikar2005]

3.2 Academic Gamification

The concept of transforming complex mathematical problems into engaging games has been successfully demonstrated by various applications. A notable example is the Flow Free game [LLC12], which demonstrates how graph connectivity problems can be transformed into intuitive puzzle mechanics:

- Release Date: 2012
- Success: Over 100 million downloads
- Core Mechanics: Connect all pairs of dots with the same color without intersections

- Algorithmic Foundation: Based on graph connectivity problems
- Educational Value: Teaches players about graph theory concepts through intuitive gameplay

This success demonstrates the potential for serious games to make complex mathematical concepts accessible to a broad audience while maintaining educational value. The Flow Free example shows that mathematical problems can be successfully gamified without losing their core algorithmic complexity, providing a model for the Alliance Divider project.

4 Game Design and Implementation

4.1 Game Design and Mechanics

4.1.1 Alliance Divider: From “Links” to “Alliances”

The Alliance Divider game transforms the abstract mathematical problem into an intuitive political strategy game:

Setting

- Multiple city-states with territories and complex political relationships
- Some relationships are friendly, others hostile
- City-states may form alliances based on their connections

Player Role

- Royal Strategist analyzing political networks
- Discover hidden alliance patterns from relationship dynamics
- Make strategic decisions to optimize alliance formation

Goal

- Cut hostile ties while preserving friendly alliances
- Form stable alliance clusters
- Minimize the total cost of cuts while maintaining valid partitions

4.1.2 Game Interface and HUD

The game provides an intuitive interface with the following elements:

- Level Display: Current level information and progress tracking
- Cut Limit Display: Remaining cuts available to the player
- Cost Display: Current total cost of performed cuts
- Territory Display: Visual representation of alliance clusters
- Hint Function: Access to optimal solution hints
- Revert Function: Undo last action for experimentation

4.2 Technical Implementation

4.2.1 Game Scene Generation and Optimization

City-State Position Generation with Poisson Disk Sampling

- Principle: Maintain a minimum distance around each point to ensure visual clarity [Bri07]
- Function: Achieve random distribution while avoiding overlapping, ensuring visual uniformity
- Benefits: Creates natural-looking city-state layouts that are easy to interpret

Connection Generation via Delaunay Triangulation

- Principle: Triangles whose circumcircles contain no other points [Del34]
- Function: Generate optimal triangular meshes, avoid thin triangles and line segment intersections
- Advantages: Creates a natural network of connections between city-states

Automatic Centering and Scaling

- Principle: Calculate the bounding box and normalize it to the screen range
- Function: Automatically adjust position and size to fit various screen resolutions
- Implementation: Ensures consistent gameplay experience across different devices

4.2.2 Third-Party TileMap Generator Integration

The game incorporates advanced tilemap generation systems to create diverse and visually appealing game environments, enhancing the overall user experience.

4.2.3 Solving the Multicut Problem: ILP + Gurobi

The core optimization engine utilizes:

- Integer Linear Programming (ILP): Mathematical formulation of the optimization problem [Sch03]
- Gurobi Solver: High-performance optimization solver for finding optimal solutions [Gur23]
- Real-time Processing: Efficient algorithms for interactive gameplay

4.2.4 System Integration: Unity to Python Cross-Language Integration

Architecture Overview

- C# Process Class: Launches Python process for optimization tasks
- JSON File Exchange: Medium for data transfer between Unity and Python
- Input Data: Graph structure and edge weight information
- Output Data: Cut edges and optimal cost calculations

Benefits of Cross-Language Integration

- Unity: Provides robust game engine capabilities and user interface [Tec23]
- Python: Offers extensive mathematical and optimization libraries
- Modularity: Separates game logic from optimization algorithms

4.3 Gamification Design

4.3.1 Alliance Territory Coloring

The game employs sophisticated visual design to represent alliance clusters:

- Color coding: Different colors represent distinct alliance groups
- Dynamic updates: Real-time visual feedback as alliances change
- Accessibility: Color-blind friendly design considerations

4.3.2 Weight and Difficulty System Architecture

The game implements a comprehensive difficulty progression system:

- Edge weights: Represent the cost of cutting relationships
- Dynamic difficulty: Adjusts complexity based on player performance
- Learning curve: Gradual introduction of complex concepts

4.4 Future Prospects and Enhancements

4.4.1 Performance Optimization

- Optimize territory display performance and resolve lag issues
- Implement efficient rendering algorithms for large-scale maps
- Enhance real-time visual feedback systems

4.4.2 Advanced Gameplay Features

- Random events: Dynamic changes to edge weights during gameplay
- Hostile city-states: Active repair of cut edges for confrontation
- Environmental factors: Mountains, rivers, and forests modify edge weights
- Multiplayer support: Collaborative and competitive gameplay modes

4.4.3 Educational Enhancements

- Tutorial system: Interactive learning modules for graph theory concepts
- Problem generation: Algorithmic creation of diverse problem instances
- Analytics dashboard: Detailed performance metrics and learning analytics

4.5 Conclusion

The Multicut Game project successfully demonstrates the potential of gamification in educational contexts, particularly for complex mathematical concepts. By combining Unity's powerful game engine with Python's optimization capabilities, the project creates an engaging platform for learning graph theory and combinatorial optimization. The modular architecture allows for future enhancements and scalability, making it a valuable tool for both education and research in algorithmic game theory.

Bibliography

[Bri07] Robert Bridson. “Fast Poisson disk sampling in arbitrary dimensions”. In: *ACM SIGGRAPH 2007 sketches* (2007), p. 22.

[CKR00] Gruia Calinescu, Howard Karloff, and Yuval Rabani. “An improved approximation algorithm for MULTICUT”. In: *Journal of Computer and System Sciences* 60.3 (2000), pp. 564–574.

[Del34] Boris Delaunay. “Sur la sphère vide”. French. In: *Bulletin de l’Académie des Sciences de l’URSS* 6 (1934), pp. 793–800.

[GHR04] Anupam Gupta, MohammadTaghi Hajiaghayi, and Harald Räcke. “Oblivious network design”. In: *Proceedings of the 36th Annual ACM Symposium on Theory of Computing* (2004), pp. 383–392.

[Gur23] LLC Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2023. URL: https://www.gurobi.com/documentation/.

[GVY97] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. “Primal-dual approximation algorithms for integral flow and multicut in trees”. In: *Algorithmica* 18.1 (1997), pp. 3–20.

[LLC12] Big Duck Games LLC. “Flow Free”. In: *Mobile Game* (2012). Over 100 million downloads.

[Sch03] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Vol. A. Algorithms and Combinatorics. Berlin: Springer-Verlag, 2003.

[Tec23] Unity Technologies. *Unity Game Engine Documentation*. 2023. URL: https://docs.unity3d.com/.