SOLVING THE HAMILTONIAN CYCLE PROBLEM WITH SAT TECHNIQUES

Term Project Proposal

Presenter：江冠緯，Student ID：B11009003

Agenda

1. Introduction
2. Problem Definition
3. SAT Reduction Approach
4. Demo

1. Introduction

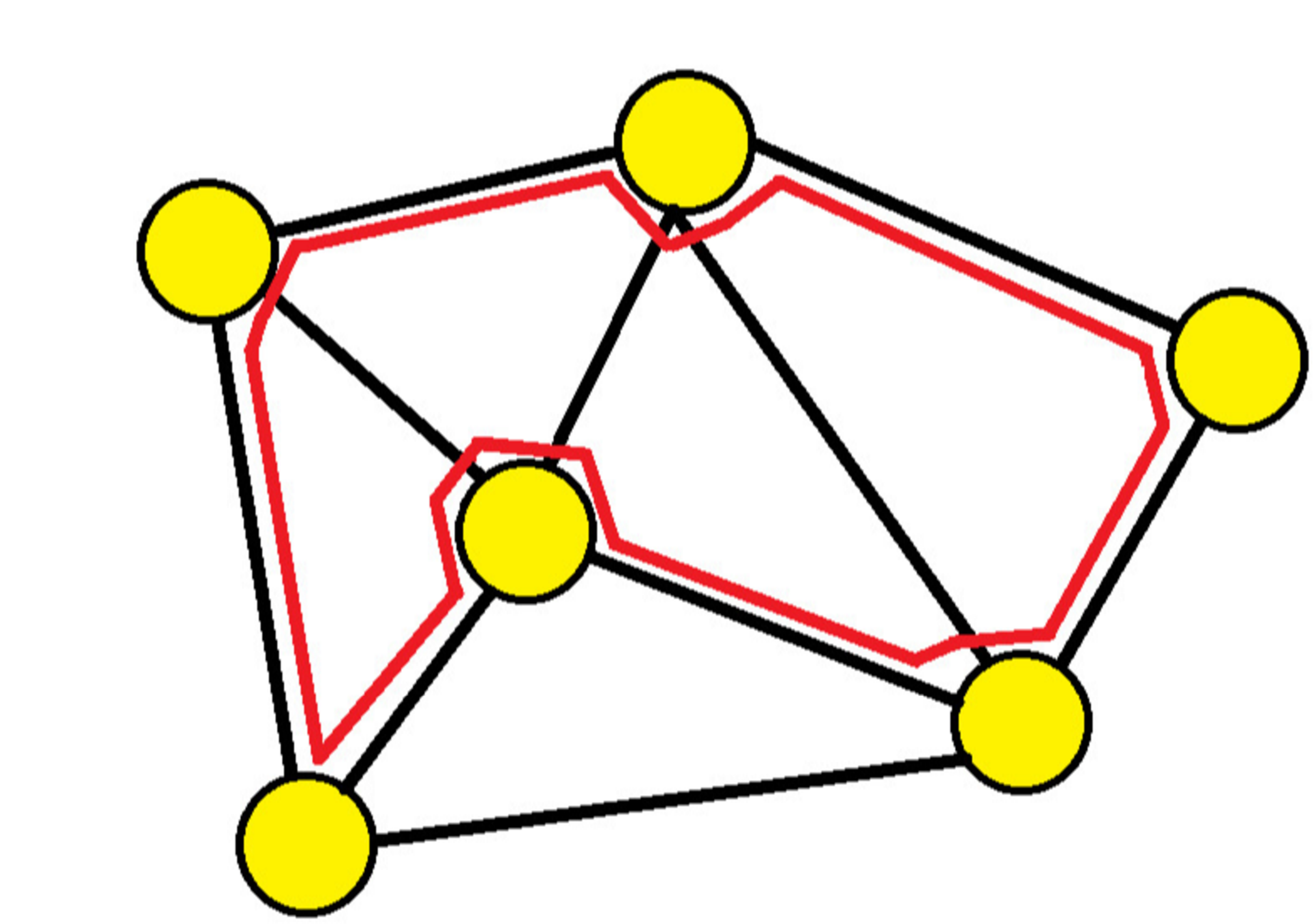
Overview

- Objective:
  - Use SAT techniques to solve the Hamiltonian Cycle Problem.
- Importance:
  - A **classic NP-Complete problem** with applications in graph theory, optimization, and circuit design.
- Approach:
  - Propose a **polynomial-time** reduction to SAT, leveraging modern SAT solvers for efficient computation.

2. Problem Definition

Hamiltonan Cycle

- Definition: Given an **undirected graph (G = (V, E))**, determine if there exists a Hamiltonian Cycle—a cycle that **visits each vertex exactly once and returns to the starting vertex**.



3. SAT Reduction Approach

Key Steps

Step 01

Define Boolean variables

represent vertex positions in the cycle

Step 02

Generate clauses

enforce constraints (e.g., each position has one vertex, each vertex appears once, adjacent vertices are connected).

Step 03

Solve the resulting CNF

Solve the resulting CNF formula using a SAT solver

Step 04

Determine Outcome

If SAT, extract the Hamiltonian Cycle; if UNSAT, no cycle exists.

Variable Definition

- Variables: For a graph with (|V| = n), define ($n \times n$) Boolean variables:

$$x_{v,p} = \{\text{True}, \quad \text{if vertex } v \text{ is at position } p \text{ in the cycle False}, \quad \text{otherwise}\}$$

- Total Variables: (n^2) (one for each vertex-position pair).
- Purpose: Encode the position of each vertex in the cycle.

Clause Generation

- Position-Cover： Each position (p) has at least one vertex

$$(x_{1,p} \lor x_{2,p} \lor \cdots \lor x_{n,p}), \quad p = 1, \ldots, n$$

- Position-Uniqueness: Each position (p) has at most one vertex:

$$\neg x_{u,p} \lor \neg x_{v,p}, \quad \forall u \neq v, p = 1, \ldots, n$$

- Purpose: Ensure each position in the cycle is occupied by exactly one vertex.

Clause Generation

- Vertex-Cover: Each vertex (v) appears in at least one position:

$$(x_{v,1} \vee x_{v,2} \vee \cdots \vee x_{v,n}), \quad v = 1, \ldots, n$$

- Vertex-Uniqueness: Each vertex (v) appears in at most one position:

$$\neg x_{v,i} \vee \neg x_{v,j}, \quad \forall i \neq j, v = 1, \ldots, n$$

- Purpose: Ensure each vertex is used exactly once in the cycle.

Clause Generation

- Adjacency-Constraints: Adjacent positions in the cycle must correspond to connected vertices:
  - For positions (p=1,...,n-1):

$$[\neg x_{u,p} \vee \neg x_{v,p+1}, \quad \forall (u,v) \notin E]$$

  - For the cycle’s return (position (n) to 1):

$$[\neg x_{u,n} \vee \neg x_{v,1}, \quad \forall (u,v) \notin E]$$

- Purpose: Ensure the cycle respects the graph’s edge structure.

DEMO