

Hamiltonian Cycle 問題的 SAT 還原與實作

Term Project Proposal

江冠緯 學號：B11009003

目录

1 摘要 (Abstract)	3
2 問題定義 (Problem Statement)	3
3 Reduction 概述 (Overview of the Reduction)	3
3.1 變數定義	3
3.2 子句生成	3
4 實作計畫 (Implementation Plan)	4
5 測試實例 (Test Instances)	5
5.1 實例 1: 8-頂點圖	5
5.2 實例 2: 9-頂點圖	5
5.3 實例 3: 10-頂點圖	6
5.4 實例 4: 8-頂點圖	7
6 參考文獻 (References)	7

1 摘要 (Abstract)

本提案旨在使用 SAT 技術求解 Hamiltonian Cycle 問題。Hamiltonian Cycle 問題：給定無向圖 $G = (V, E)$ ，判斷是否存在一條環路訪問每個頂點恰好一次且返回起點。我們將此問題轉成 SAT，生成 CNF 公式且使用現代求解器進行求解。

2 問題定義 (Problem Statement)

Hamiltonian Cycle 問題是一個經典的 NP-Complete 問題。

- **輸入：** 圖 $G = (V, E)$ ，其中 V 是頂點集合、 E 是邊集合。
- **問題：** 是否存在一條 Hamiltonian 環，恰好訪問每個頂點一次且返回起點

3 Reduction 概述 (Overview of the Reduction)

本節闡述如何將 Hamiltonian Cycle 問題多項式時間地還原 SAT。

3.1 變數定義

令 $|V| = n$ 。對每個頂點 $v \in V$ 及每個位置 $p \in \{1, \dots, n\}$ 定義布林變數

$$x_{v,p} = \begin{cases} \top, & \text{若頂點 } v \text{ 位於環的第 } p \text{ 個位置;} \\ \perp, & \text{否則。} \end{cases}$$

共使用 $n \times n$ 個變數。

3.2 子句生成

本還原共生成以下五類子句：

1. **Position-Cover:** 每個位置至少有一個頂點

$$(x_{1,p} \vee x_{2,p} \vee \dots \vee x_{n,p}), \quad p = 1, \dots, n.$$

2. **Position-Uniqueness:** 每個位置至多一個頂點

$$\neg x_{u,p} \vee \neg x_{v,p}, \quad \forall u \neq v, \quad p = 1, \dots, n.$$

3. **Vertex-Cover:** 每個頂點至少出現在一個位置

$$(x_{v,1} \vee x_{v,2} \vee \dots \vee x_{v,n}), \quad v = 1, \dots, n.$$

4. **Vertex-Uniqueness:** 每個頂點至多出現在一個位置

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

5. **Adjacency-Constraints:** 相鄰位置必須在圖中相連：

- 對所有不相鄰頂點對 $(u, v) \notin E$ 及每個 $p = 1, \dots, n - 1$, 加入

$$\neg x_{u,p} \vee \neg x_{v,p+1}.$$

- 對環回到起點：對所有 $(u, v) \notin E$, 加入

$$\neg x_{u,n} \vee \neg x_{v,1}.$$

4 實作計畫 (Implementation Plan)

- **環境：** Python 3.8+, `python-sat` (PySAT), Cadical195 SAT 求解器。
- **輸入格式：** 圖的邊列表 (`graph.txt`), 第一行是頂點數 n 和邊數 m , 後續 m 行是邊 $u\ v$ 。
- **執行流程：**
 - 執行還原與求解 script:

```
python reduce_hc.py graph.txt
```

 - script 功能:
 - * 讀取圖且生成 $n \times n$ 個布林變數 $x_{v,p}$ 。
 - * 生成五類 CNF 子句 (Position-Cover, Position-Uniqueness, Vertex-Cover, Vertex-Uniqueness, Adjacency-Constraints)。
 - * 使用 Cadical195 求解器判斷 SAT 或 UNSAT。
 - * 若 SAT, 解碼且輸出 Hamiltonian 環序列。
- **結果解釋：**
 - SAT: 輸出 Hamiltonian 環, 例如 $1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow 1$ 。
 - UNSAT: 表示無 Hamiltonian 環。
- **測試實例：**
 - 實例 1: 8-頂點圖 (`graph1.txt`)
 - 實例 2: 9-頂點圖 (`graph2.txt`)
 - 實例 3: 10-頂點圖 (`graph3.txt`)
 - 實例 4: 8-頂點圖 (`graph4.txt`)

5 測試實例 (Test Instances)

5.1 實例 1: 8-頂點圖

- 輸入 (`graph1.txt`):

```
8 12
1 4
4 2
2 6
6 3
3 8
8 5
5 7
7 1
1 5
2 7
3 4
6 8
```

- 輸出:

SAT

Hamiltonian Cycle: 1 → 4 → 2 → 6 → 3 → 8 → 5 → 7 → 1

- 解碼結果: Hamiltonian 環是 1 → 4 → 2 → 6 → 3 → 8 → 5 → 7 → 1。

5.2 實例 2: 9-頂點圖

- 輸入 (`graph2.txt`):

```
9 15
1 7
7 3
3 9
9 2
2 8
8 4
4 6
```

```
6 5  
5 1  
1 3  
2 4  
3 5  
4 7  
6 9  
8 1
```

- 輸出：

SAT

Hamiltonian Cycle: 1 -> 7 -> 3 -> 9 -> 2 -> 8 -> 4 -> 6 -> 5 -> 1

- 解碼結果： Hamilton 環是 1 → 7 → 3 → 9 → 2 → 8 → 4 → 6 → 5 → 1。

5.3 實例 3：10-頂點圖

- 輸入 (*graph3.txt*):

```
10 16  
1 8  
8 3  
3 10  
10 4  
4 7  
7 2  
2 9  
9 5  
5 6  
6 1  
1 4  
2 5  
3 7  
4 8  
5 10  
6 9
```

- 輸出：

SAT

Hamiltonian Cycle: 1 → 8 → 3 → 10 → 4 → 7 → 2 → 9 → 5 → 6 → 1

- **解碼結果:** Hamiltonian 環是 1 → 8 → 3 → 10 → 4 → 7 → 2 → 9 → 5 → 6 → 1。

5.4 實例 4: 8-頂點圖

- 輸入 (graph4.txt):

```
8 14
1 6
6 2
2 8
8 3
3 5
5 4
4 7
7 1
1 3
2 4
3 6
4 8
5 7
6 1
```

- 輸出:

SAT

Hamiltonian Cycle: 1 → 6 → 2 → 8 → 3 → 5 → 4 → 7 → 1

- **解碼結果:** Hamiltonian 環是 1 → 6 → 2 → 8 → 3 → 5 → 4 → 7 → 1。

6 參考文獻 (References)

1. OpenDSA. 3-SAT to Hamiltonian Cycle reduction.