# Řešení Intersection Number pomocí SAT solveru

## Přesný popis problému

**Definice Intersection Number**

Intersection Number grafu je definováno jako minimální počet intervalů , takových že:

* Každému vrcholu lze přiřadit množinu intervalů .
* Každá hrana (má alespoň jeden průnik mezi intervaly a , tj. .

**Praktická aplikace**

Intersection Number má aplikace v analýze sítí, kde je třeba minimalizovat počet zdrojů (např. kanálů, procesorů) potřebných ke komunikaci mezi vrcholy.

## Strategie řešení problému

Řešení se skládá z následujících kroků:

1. **Redukce na SAT**: Transformace problému Intersection Number na splnitelnost SAT formule v CNF formátu.
   * Definujeme logické proměnné , které označují, zda vrchol je přiřazen k intervalu .
   * Vyjádříme omezení problému pomocí klauzulí.
   * Postupně testujeme různé hodnoty (počtu intervalů), dokud nenajdeme minimální , které má splnitelné řešení.
2. **Generování DIMACS CNF formátu**:
   * Výstupem je soubor obsahující logické proměnné a klauzule ve standardním formátu SAT solveru.
3. **Použití SAT solveru Glucose**:
   * SAT solver testuje splnitelnost formule.
   * Pokud je splnitelná, dekódujeme řešení zpět na množiny intervalů pro jednotlivé vrcholy.
4. **Optimalizace** :
   * Postupujeme iterativně od malých hodnot kkk směrem nahoru, dokud nenajdeme nejmenší splnitelné řešení.

## Podrobný návrh zakódování problému

**Proměnné**

Každý vrchol má přiřazenou množinu proměnných  kde:

* , pokud vrchol je přiřazen intervalu .
* , jinak.

Celkem tedy máme proměnných.

**Omezení (constrainty)**

1. **Pokrytí hran**: Pro každou hranu musí existovat alespoň jeden interval , kde.
   * To můžeme vyjádřit v CNF:

*To znamená, že pro každou hranu máme klauzulí, kde je počet intervalů.*

1. **Optimalizace počtu intervalů**: Nejmenší , které splňuje všechna omezení, najdeme iterativním zvyšováním .
2. **Povolené hodnoty proměnných**: Proměnné ​ mohou nabývat pouze hodnot nebo .

## Implementace v Pythonu

**Podrobnosti implementace**

Následující části pokryjí:

1. Vstupní graf a počet intervalů.
2. Generování CNF formule a zápis do DIMACS.
3. Spuštění Glucose solveru a zpracování výstupu.
4. Iterativní optimalizaci .

## Optimalizace a testování

**Optimalizace**

Spusťte SAT solver pro různé hodnoty (např. )a najděte nejmenší splnitelné .

**Testovací instance**

1. **Malý splnitelný graf**: Např. trojúhelník s .
2. **Malý nesplnitelný graf**: Např. s .
3. **Větší grafy**: Např. nebo mřížkové grafy.

**Závěr**

Tento projekt implementuje řešení problému Intersection Number grafu pomocí SAT solveru Glucose. Hlavním cílem bylo vytvořit skript, který dokáže:

1. **Efektivně zakódovat problém Intersection Number do SAT**:
   * Transformace probíhá pomocí logických proměnných a klauzulí v CNF formátu, což umožňuje využití standardních SAT solverů.
2. **Optimalizovat počet intervalů kkk**:
   * Skript iterativně hledá minimální kkk, pro které je problém splnitelný, čímž poskytuje optimální řešení.
3. **Dekódovat řešení SAT solveru**:
   * Výstup SAT solveru je převeden zpět na intervalové přiřazení pro vrcholy grafu, což umožňuje snadnou interpretaci výsledků.

**Experimentální výsledky**

Byly otestovány různé instance:

* Malé splnitelné a nesplnitelné instance sloužily k ověření správnosti implementace.
* Středně velké instance umožnily ověřit výkonnost SAT solveru a identifikovat limity, např. u grafů s více než několika desítkami vrcholů a hran.

**Možné rozšíření**

1. **Efektivnější redukce**:
   * Zkoumat alternativní zakódování CNF pro zlepšení výkonu solveru.
2. **Paralelní vyhledávání kkk**:
   * Paralelizace testování hodnot kkk by mohla zrychlit nalezení optimálního řešení.
3. **Rozšíření na další problémy**:
   * Metodika použitá v tomto skriptu může být snadno přizpůsobena k řešení jiných problémů z oblasti teorie grafů, jako jsou chromatičnost nebo klikové pokrytí.

**Závěrečné hodnocení**

Projekt demonstruje sílu SAT solverů při řešení složitých kombinatorických problémů. Navržený skript je flexibilní, uživatelsky přívětivý a dostatečně efektivní pro menší až středně velké instance. Implementace poskytuje solidní základ pro další experimenty a výzkum v oblasti aplikací SAT solverů v teorii grafů.