SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

VÝSKUM RIEŠENIA RIEDKYCH MRHS ROVNÍC
DOKUMENTĆIA

2021 tím číslo 15

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

VÝSKUM RIEŠENIA RIEDKYCH MRHS ROVNÍC
DOKUMENTĆIA

Študijný program:	Aplikovaná informatika
Predmet:	TP – Tímový projekt
Prednášajúci:	Ing. Eugen Antal, PhD.
Cvičiaci:	prof. Ing. Pavol Zajac, PhD.

Bratislava 2021 **tím číslo 15**

Obsah

Zoznam obrázkov a tabuliek

Zoznam algoritmov

Zoznam výpisov

1 O projekte

1.1 Predstavenie tímu

Na tomto projekte budú spolupracovať nasledovní študenti FEI STU.

1.1.1 Jakub Kupkovič

Člen Jakub Kupkovič pracoval na bakalárskej práci kde implementoval software na riešenie MRHS rovníc na grafickej karte. Má skúsenosti s programovaním v Pythone, C++, Jave a implementáciou riešení v CUDA prostredí. Tému si vybral kvôli záujmu vzdelávania v danej oblasti. V tíme bude mať pozíciu vývojára.

1.1.2 Daniel Janči

Ďalší člen tímu, Daniel Janči, sa zaoberal výskumom v oblasti kryptografie. V záverečnej práci bakalárskeho štúdia sa zaoberal problémom SAT a využíval SAT solvery pre výskumné účely. Skúsenosti má prevažne v jazyku Java, Python a je vždy ochotný učiť sa nové veci. V tíme bude pracovať prevažne na vývoji.

1.1.3 Jakub Lengvarský

Člen Jakub Lengvarský má skúsenosti s vývojom webových aplikácií. Vývojom jednej z nich sa venoval aj vo svojej bakalárskej práci. Skúsenosti má najmä v jazykoch Java, PHP, Javascript a Python. Zaujíma ho aj problematika počítačovej bezpečnosti. V tíme bude mať na starosti tvorbu dokumentácie, správu webovej stránky a čiastočne aj samotný vývoj.

1.1.4 Jozef Genšor

Posledným členom tímu je Jozef Genšor. V bakalárskej práci sa zaoberal detekciou voľných parkovacích miest z kamerového záznamu. Skúsenosti či už pracovné alebo v osobných projektoch má hlavne v jazykoch Python a PHP. Tento projekt ho oslovil z dôvodu rozšírenia si vedomostí v problematike počítačovej bezpečnosti. V tomto projekte bude mať na starosti repozitár projektu a jeho kompozíciu a v neposlednom rade aj pomoc pri samotnom vývoji.

1.2 Motivácia

Šifrovanie má v dnešnej dobe široké využite. Existuje niekoľko algoritmov ktoré zabezpečujú bezpečné zašifrovanie informácie. Účinnosť takéhoto algoritmu vieme zistiť pomocou kryptoanalýzy. Pri symetrických šifrách. Ak je algoritmus slabý tak vieme rýchlejšie zistiť kľuč k danej šifre. Algebraická kryptoanalýza, konkrétne reprezentácia pomocou systému rovníc s viacerými pravými stranami (MRHS) nie je v dnešnej rozšírená. Našou prácou by

sme chceli túto skutočnosť zmeniť a vytvoriť prostredie, ktoré uľahčí používateľom prácu.

1.3 Návrh

Náš tím sa pokúsi vytvoriť prostredie s prostriedkami na jednoduchú implementáciu rôznych funkcií riešenia MRHS. Toto zabezpečíme výskumom danej problematiky a implementáciou optimálnych riešení v tvare Python knižnice. Zameriame sa na užitočné funkcie ako je napríklad načítanie matíc, ich riešenie pomocou rôznych metód. Všetka funkcionalita bude vo forme dokumentácie nahraná na web.

1.4 Časový plán

Meno	Po	Ut	St	Št	Pi	So	Ne
Jakub Kupkovič	2h	2h		2h		2h	
Daniel Jančí	2h	2h		2h		2h	
Jakub Lengvarský	2h	2h		2h		2h	
Jozef Genšor	2h	2h		2h		2h	

1.5 Prostriedky

Pre širšiu dostupnosť implementácie pre rozličné hardwary bude potrebné zaobstarať tieto komponenty kvôli následnému testovaniu. Bude sa jednať najme o grafické karty od rozličných výrobcov alebo s rozličnými architektúrami.

2 MRHS rovnica

Táto dokumentácia sa zaoberá rovniciam s viacerými pravými stranami a ich riešeniami (MRHS).

Označme si konečné pole s dvoma prvkami ako $\boldsymbol{F}$. Všetky vektory nad $\boldsymbol{F}$ sú riadkové vektory a sú označené malými písmenami. Množiny vektorov sú označené veľkými písmenami a všetky matice sú označené tučným písmom.

Definícia 1. *Rovnica s viacerými pravými stranami, je výraz v tvare*

$$x\boldsymbol{M} \in S,$$

kde $\boldsymbol{M}$ je matica $n \times l$ a $S \subset F^l$ je množina l-bitových vektorov. Hovoríme, že $x \in F^n$ je riešením MRHS rovnice vtedy, a len vtedy, ak $x\boldsymbol{M} \in S$.

Sústava MRHS rovníc M, je množina m MRHS rovníc, s rovnakým rozmerom n.

$$M = \{x\boldsymbol{M}_i \in S_i | 1 \leq i \leq m\}$$

kde každé $\boldsymbol{M}_i$ je matica $(n \times l_i)$ a $S_i \supset F^{l_i}$. Vektor $x \in F^n$ je riešením MRHS sústavy M, ak je riešením pre všetky MRHS rovnice v M.

***Zjednotená sústava matíc**: Vzhľadom na MRHS sústavu rovníc $M = \{x\boldsymbol{M}_i \in S_i\}$ môžeme spojiť všetky matice $\boldsymbol{M}_i$, pretože všetky majú rovnaký počet riadkov. Takto spojenú maticu označujeme $\boldsymbol{M}$, a nazývame ju zjednotená sústava matíc.*

$$\boldsymbol{M} = [\boldsymbol{M}_i | \boldsymbol{M}_i | ... | \boldsymbol{M}_m]$$

Podobne budeme označovať aj $S_1 \times S_2 \times ... \times S_m$ ako S. Riešením sústavy MRHS je nájdenie takého $x \in F^n$, pre ktoré platí $x\boldsymbol{M} \in S$.

2.1 Riedkosť matice

Zjednotená sústava matíc je dvojrozmerný dátový objekt tvorený m riadkami a n stĺpcami a preto obsahuje m × n prvkov. Počet prvkov s nulovou hodnotou vydelený celkovým počtom prvkov sa nazýva riedkosť (sparsity) matice. Ak je riedkosť matice väčšia ako 0.5, potom sa matica nazýva riedka.

3 Implementácia riešenia MRHS rovníc

Táto implementácia riešenia MRHS rovníc bolo implementovaná v jazyku Python (konkrétne verzia 3.8). Medzi základné funkcionality programu patrí generovanie a inicializácia matice s náhodnými hodnotami, načítanie matice zo súboru a nájdenie všetkých riešení MRHS rovnice.

4 Trieda MRHS

Hlavná trieda knižnice. Uchováva v sebe údaje o sústave Matíc. Štrukturovaná je po blokoch, kde každý blok má úloženú hlavnú maticovú časť a časť s právými stranami.

4.1 Konštruktor ___init___(vector_size)

Inicializuje MRHS maticu s počtom riadkov.

- vector_size - Počet riadkov sústavy

4.2 Premenné

- blocks_array - Obsahuje list všetkých blokov
- vector_size - Obsahuje počet riadkov

4.3 Funkcia init_with_matrix(vectors)

Funkcia vytvorí MRHS pomocou listu, vo formáte listu blokov, kde 1 blok je reprezentovaný ako 2 listy, ktoré obsahujú riadky v hlavnej matici a odpovede.

- vectors - 4D matica [Bloky[Matica[Vektory],RHS[Vektory]]]

4.4 Funkcia generate_random_block_array(blocks,rhs_fill,seed=

Funkcia vygeneruje náhodné hodnoty matice.

- blocks - pole reprezentuje veľkosti blokov
- rhs_fill - reprezentuje percento pravých strán

4.5 Funkcia print_mrhs()

Funkcia vypíše maticu do konzoly.

```
100 0 1 100
010 0 1 010
000 1 1 001
000 0 0 100
001 0 1 111
------------------------------
111 0 1 101
101    010
```

Listing 1: Výpis MRHS

4.6 Funkcia find_all_solutions()

Nájde všetky riešenia MRHS sústavy a vráti ich v tvare poľa.

4.6.1 Navratová hodnota

Pole obsahujúce všetky vektory, ktoré sú riešením MRHS sústavy.

4.7 Generator find_solution()

Generátor vracia riešenia MRHS sústavy. Ak už nemá riešenie tak vráti hodnotu None.

5 Trieda CreateFile(MRHS)

Táto trieda slúži na vypísanie MRHS v štandardnom formáte. Pre vytvorenie objektu tohto typu je potrebné na vstup dať už vytvorený objekt typu MRHS. Trieda CreateFile obsahuje jednu funkciu s názvom create_file.

5.1 Funkcia create_file(file_name)

Vstup tejto funkcie, file_name, je reťazec, ktorý predstavuje názov textového súboru (aj s príponou .txt). Táto funkcia po zavolaní vytvorí textový súbor s názvom file_name a vypíše do neho MRHS sústavu z daného objektu typu CreateFile.

```
from MRHS_Solver import CreateFile as cf

file = cf.CreateFile(mrhs)
file.create_file('output.txt')
```

Listing 2: Príklad volania funkcie create_file(file_name)

```
5
4
3 2
1 1
1 1
3 2
[1 0 0  0  1  1 0 0]
[0 1 0  0  1  0 1 0]
[0 0 0  1  1  0 0 1]
[0 0 0  0  0  1 0 0]
[0 0 1  0  1  1 1 1]
[1 1 1]
[1 0 1]

[0]

[1]

[1 0 1]
[0 1 0]
```

Listing 3: Príklad výstupného textového súboru funkcie create_file(file_name)

6 Trieda LoadFile(MRHS)

Táto trieda slúži na načítanie MRHS zo štandardného formátu. Ako vstupný parameter pre vytvorenie inštancie tohto objektu je potrebné zadať cestu k súboru, ktorý obsahuje MRHS v štandardnom formáte. Trieda LoadFile obsahuje dve interné funkcie pre načítanie vstupnej matice a riešení a funkciu get_final_matrix ktorá vráti riadky a im zodpovedajúce riešenia v blokoch.

```
5 --> Pocet Riadkov
4 --> Pocet Stlpcov
3 2 -- [
1 1
1 1
3 2 -- ] --> BlokN : Pocet Riadkov, BlokN : Pocet Rieseni
[1 0 0  0  1  1 0 0] -- [
[0 1 0  0  1  0 1 0]
[0 0 0  1  1  0 0 1]
[0 0 0  0  0  1 0 0]
[0 0 1  0  1  1 1 1] -- ] --> Matica
[1 1 1] -- [
[1 0 1]

[0]

[1]

[1 0 1]
[0 1 0] -- ] --> Riesenia pre BlokN
```

Listing 4: Príklad vstupného textového súboru triedy LoadFile(file_name)

6.1 Funkcia get_final_matrix()

Táto funkcia vráti vstup vo formáte blokov ktoré sú reprezentované ako pole. Každý index v poli zodpovedá jednému bloku, ktorý obsahuje dve vnorené polia, ktoré zodpovedajú riadkom bloku a jeho riešeniam.

```
from MRHS_Solver import LoadFile as lf

mat = lf.LoadFile("input.txt").get_final_matrix()
```

Listing 5: Príklad volania funkcie get_final_matrix()

```
mat = [
        [
            [
              [1, 0, 0],[0, 1, 0],[0, 0, 0],[0, 0, 0],[0, 0, 1] #Hlavna matica
            ]
            ,
            [
              [1, 1, 1],[1, 0, 1] #RHS
            ]
        ], #BLOCK
        [
            [
                [0], [0], [1], [0], [0] #Hlavna matica
            ]
            ,
            [
                [0] #RHS
            ]
        ], #Block
        [
            [
                [1], [1], [1], [0], [1] #Hlavna matica
            ]
            ,
            [
                [1] #RHS
            ]
        ], #Block
        [
            [
              [1, 0, 0],[0, 1, 0],[0, 0, 1],[1, 0, 0],[1, 1, 1] #Hlavna matica
            ]
            ,
            [
              [1, 0, 1],[0, 1, 0] #RHS
            ]
        ] #Block
    ]
```

Listing 6: Príklad výstupu funckie get_final_matrix() pre vstupný súbor Listing 4.

Prílohy