



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**DEMONSTRACE - KARGERŮV ALGORITMUS**

DEMONSTRATION - KARGER'S ALGORITHM

**PROJEKTOVÁ DOKUMENTACE**

PROJECT DOCUMENTATION

**AUTOR PRÁCE**

AUTHOR

**KATEŘINA PILÁTOVÁ, MICHAL TABÁŠEK**

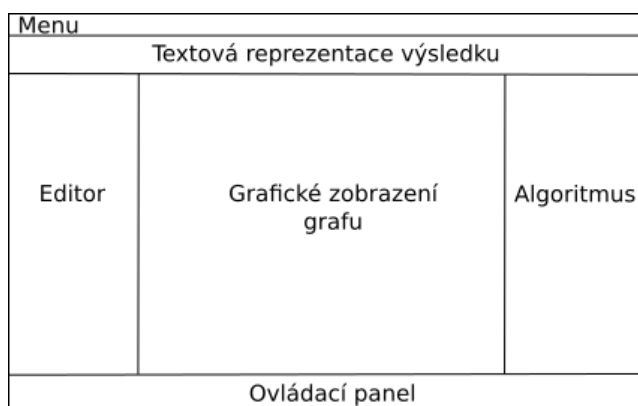
**BRNO 2017**

## Úvod

Cílem tohoto projektu je návrh a implementace demonstrační aplikace Kargerova algoritmu pro nalezení minimálního řezu. Daný algoritmus byl navržen a implementován s využitím vhodných doporučených nástrojů.

## Návrh aplikace

Předběžný návrh aplikace je důležitým krokem ve vývoji. Při tomto postupu je možno se vyhnout případným problémům, které by mohly nastat v pozdější fázi vývoje. Prvním krokem byl návrh základního rozložení hlavního okna aplikace.



Obrázek 1: Uspořádání hlavního okna aplikace.

## Použité nástroje

Dalším krokem návrhu aplikace byl výběr programovacího jazyka, ve kterém bude aplikace napsána. Výsledná aplikace by měla mít grafické rozhraní (GUI) a měla by být spustitelná na jakékoliv platformě, především na studentském serveru Merlin. Na základě těchto kritérií byl zvolen jazyk Java spolu s knihovnou Swing pro tvorbu grafického rozhraní. Ze seznamu povolených formátů grafů byl vybrán formát mxGraph a knihovna JGraphX s tímto formátem pracující.

### Swing

Swing [2] je knihovna plně založená na platformě Java a postavená na AWT (Abstract Windowing Toolkit), sloužící k ovládání počítače pomocí grafického rozhraní. S využitím této knihovny je možno vytvářet okna (JFrame), dialogy (JOptionPane), tlačítka (JButton), seznamy a mnoho dalšího.

### JGraphX

JGraphX [1] je knihovna licencovaná pod licencí BSD, založená na Java Swing knihovně, poskytující funkcionalitu pro vizualizaci a interakci s grafy založenými na systému uzelná. JGraphX také obsahuje funkce pro podporu XML šablon, různých importů a exportů a rozvržení grafu.

## Implementace

Hlavní částí vývoje byla implementace algoritmu podle zvolené varianty zadání, v případě tohoto projektu o Kargerův algoritmus.

### Algoritmus

Jedná se o deterministický randomizovaný algoritmus, který náhodně vybírá hranu mezi všemi hranami grafu a slučuje koncové uzly této hrany, dokud nezůstanou pouze dva tzv. super uzly [3] reprezentující dvě skupiny uzlů v grafu a jedna hrana/skupina hran představující velikost řezu. Jednoduchý pseudokód hlavní smyčky programu:

---

**Algorithm 1** Karger Algorithm

---

```
1: Let  $G = (V, E)$ 
2: while  $|V| > 2$  do
3:   nahodne vyber dva sousedni uzly  $v_1, v_2$ 
4:   mergeCells( $v_1, v_2$ )
```

---

Spojení dvou uzlů:

---

**Algorithm 2** *mergeCells*( $v_1, v_2$ )

---

```
1: for each  $v \in \text{Adj}[v_2]$  do
2:   presmeruj hranu  $(v, v_2)$  na  $(v, v_1)$ 
3:   pripadne multihrany oznac poctem hran nad reprezentujici hranou
4:   prejmenuj  $v_1$  // konkatence nazvu  $v_1$  a  $v_2$ 
5:   odstran  $v_2$  a vsechny jeho hrany
```

---

Časová složitost Kargerova algoritmu je polynomiální. Je potřeba vyzkoušet všechny možnosti – těch je  $|E|!$ . Pro každou možnost se spustí jeden běh o  $O(|E|)$  krocích, kde každý krok má polynomiální složitost (zejména kvůli vyhledávání hran ze seznamu při označování multihran a přesměrovávání hran).

### Běh aplikace

Na začátku je uživateli zobrazen výchozí graf. Uživatel může buď spustit krok/běh/algoritmus nad tímto grafem, nebo tento graf upravit, anebo si vytvořit svůj vlastní graf od základů.

Vnitřně je graf reprezentován třídou `KargerGraph`, kde je uložen samotný graf a informace o něm. Důležitými proměnnými jsou `graphEdges`, kde je seřazený seznam hran, `curOrder` s aktuálním pořadím výběru hran v tomto běhu, `runs` obsahující všechny zatím dosažené výsledky či `encodedResetGraph`, který uchovává původní graf, který se obnovuje při každém dalším běhu algoritmu.

V každém kroku se nejprve na základě `curOrder`, čísla kroku a seznamu hran vybere hrana, která se bude odstraňovat a podle ní se určí na ni napojené uzly, co se spojí. Uvnitř metody `mergeCells(v1, v2)` již probíhá samotné spojování (tedy přesměrovávání hran, vytváření multihran a odstraňování uzlu  $v_2$ ).

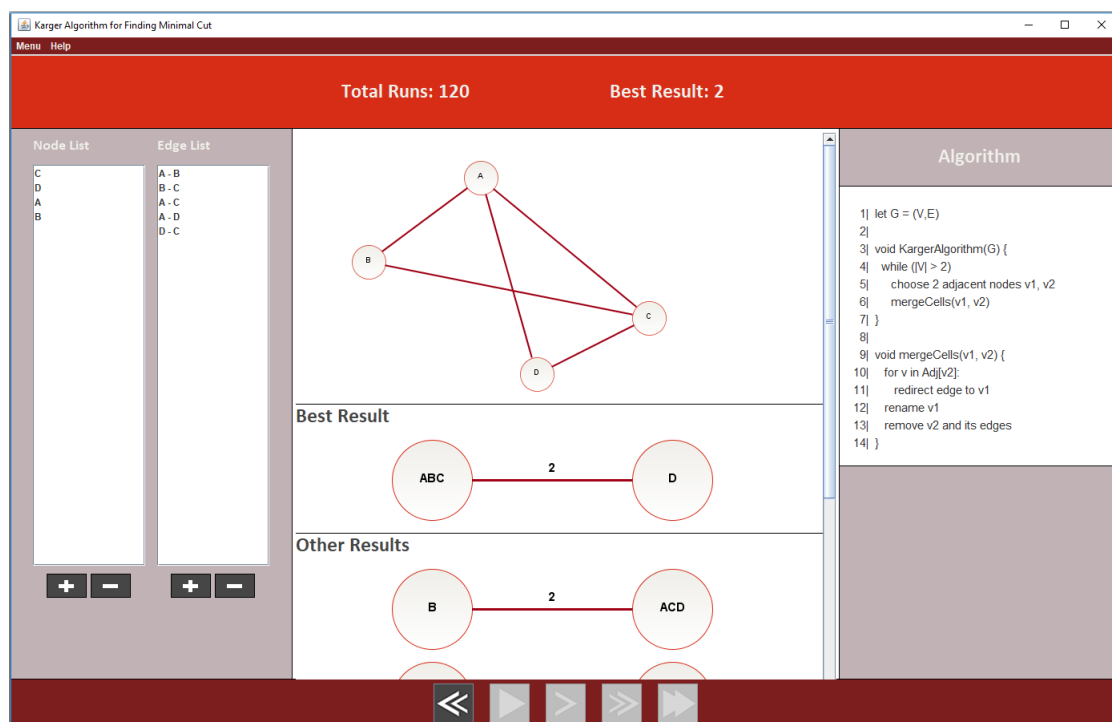
Pokud uživatel zvolí manuální krokování aplikace, je mu v pravém panelu vždy zobrazen řádek algoritmu, ve kterém se aplikace právě nachází. V první fázi tohoto kroku jsou mu

zvýrazněny ty uzly, co se spojí a po dokončení druhé fáze dostane výsledek po spojení těchto uzlů jako při normálním kroku.

Při dokončení celého běhu algoritmu je kromě původního grafu uživatel zobrazen nejlepší výsledek a ostatní výsledky jsou seřazeny od nejlepšího pod ním.

## Struktura aplikace

Celá aplikace je tvořena jedním hlavním oknem **MainWindow**. Rozložení hlavního okna je zobrazeno v návrhu aplikace na obrázku 1. Po spuštění aplikace proběhne načtení a zobrazení výchozího grafu. Uživatel má možnost vybrat pro zpracování jiný graf. Graf je možno editovat přidáním nebo odebráním uzlů a hran. Případná komunikace s uživatelem je realizována prostřednictvím dialogových oken.



Obrázek 2: Zobrazení výsledku aplikace algoritmu na daný graf.

## Aplikace

### Minimální požadavky

- OpenJDK
- Ant (kompilace a spouštění)
- JGraphX (knihovna pro práci s grafem)

## Spuštění a instalace

Odevzdaný archiv obsahuje:

- *src* – adresář se zdrojovými soubory
- *lib* – adresář s využitými knihovnami
- *latex* – zdrojové kódy této dokumentace
- *README.md*
- *dokumentace.pdf* – tato dokumentace
- *build.xml* - skript pro kompilaci a spuštění aplikace pomocí Ant

Pro spuštění aplikace je nutno mít nainstalovaný OpenJDK a Java knihovnu Apache Ant. Aplikace se spouští přes soubor *karger.jar*.

## Možnosti aplikace

- vytvoření/načtení/uložení grafu ve formátu XML
- editace grafu – přidání a odebrání uzlu/hrany
- ovládání grafu pomocí ovládacího panelu – resetování, další krok, dokončení jednoho běhu, dokončení algoritmu
- podrobnější krokování provádění algoritmu spolu s vyznačením právě provedených částí pseudokódu
- přesun uzlů pomocí kliknutí a tažení myši
- zobrazení uživatelského manuálu pod záložkou *Help* v horním menu

## Závěr

Cílem práce bylo vytvořit aplikaci, která demonstruje Kargerův algoritmus pro nalezení minimálního řezu. Samotné řešení projektu s využitím knihovny JGraphX bylo poučné a vedlo k pochopení fungování Kargerova algoritmu. Řešení výsledné aplikace by mělo uživateli pomoci minimálně s pochopením základního principu algoritmu.

# Literatura

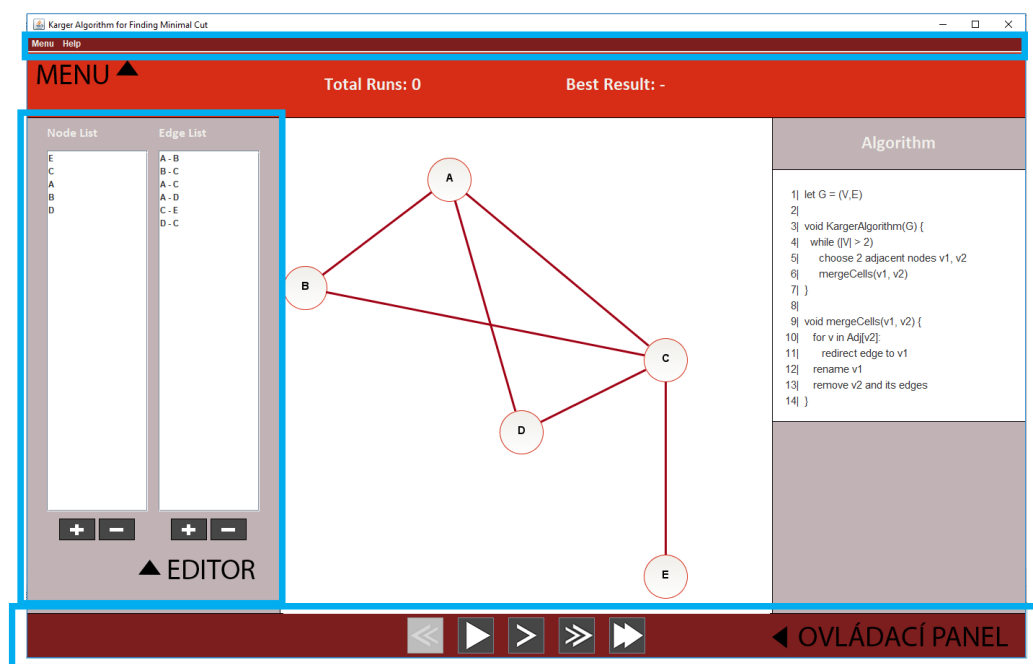
- [1] *JGraphX (JGraph 6) User Manual*. [Online; navštíveno 12.10.2017].  
URL <https://github.com/jgraph/jgraphx>
- [2] *Lesson: Getting Started with Swing*. [Online; navštíveno 12.10.2017].  
URL <https://docs.oracle.com/javase/tutorial/uiswing/start/index.html>
- [3] Williams, V. V.: *Min Cut and Karger's Algorithm*. Květen 2016, [Online; navštíveno 12.10.2017].  
URL <http://web.stanford.edu/class/archive/cs/cs161/cs161.1166/lectures/lecture15.pdf>

# Příloha A

## Uživatelská příručka

Tato aplikace slouží k demonstraci Kargerova algoritmu.

### Ovládání aplikace

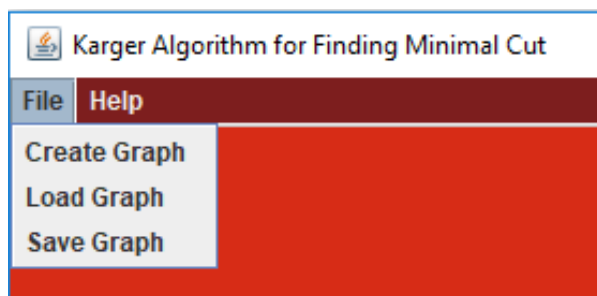


Obrázek A.1: Rozhraní ovládání aplikace.

Aplikaci lze ovládat pomocí:

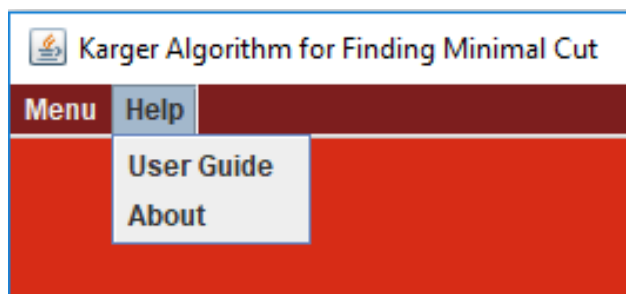
- Menu – Práce se souborem, nápověda.
- Editoru – Úprava grafu.
- Ovládacího panelu – Aplikace algoritmu na graf.

## Menu



Možnosti práce se souborem:

- vytvoření nového grafu
- načtení uloženého grafu ve formátu XML
- uložení grafu ve formátu XML





Možnosti nápovědy:

- uživatelská příručka
- o aplikaci

## Editor

Editor obsahuje seznam uzlů (*Node List*) a seznam hran (*Edge List*). Ke každému z těchto seznamů jsou přidružena dvě tlačítka:

 - Vložení uzlu/hrany do grafu

 - Odebrání uzlu/hrany z grafu

## Ovládací panel




Ovládací panel obsahuje pět různých tlačítek, přičemž aplikace může běžet ve dvou různých módech. První tlačítko zleva je:

 – Resetování grafu. Vráť graf do počátečního stavu a umožňuje začít znovu.




Zbývající čtyři tlačítka pracují v různých módech aplikace:

### **Tzv. Single Run Mode**

-  – Kliknutím na toto tlačítko dojde k provedení jednoho kroku algoritmu, tedy ke sloučení jedné dvojice uzlů a zobrazení aktualizovaného grafu.
-  – Kliknutím na toto tlačítko dojde ke spuštění jednoho běhu algoritmu a zobrazení výsledku tohoto běhu.
-  – Kliknutím na toto tlačítko je aktivováno manuální krokování algoritmu v pravém panelu, doprovázeno zobrazováním změn na grafu.

### **Multiple Runs Mode**

-  – Kliknutím na dané tlačítko je spuštěn kompletní běh algoritmu.

Algoritmus tedy proběhne pro všechny možné varianty. Po dokončení zobrazí graf v počátečním stavu, nejlepší možný výsledek a všechny další výsledky.