

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Generování sítí**

## **Networks Generation**

2017

Jakub Uhlík

## Zadání bakalářské práce

Student: **Jakub Uhlík**  
Studijní program: B2647 Informační a komunikační technologie  
Studijní obor: 2612R025 Informatika a výpočetní technika  
Téma: **Generování sítí**  
**Networks Generation**

Jazyk vypracování: čeština

### Zásady pro vypracování:

S mnohými reálnými datovými kolekcemi, které představují svět kolem nás (např. Web, Internet, sociální sítě, ...), můžeme pracovat jako se sítěmi (grafy), tedy jako s entitami spojenými vazbami. Z analýzy takovýchto dat jsou známy různé jejich vlastnosti. Cílem práce je implementace algoritmů pro generování datových kolekcí (sítí, grafů) majících různé požadované vlastnosti.

1. Seznamte se s tzv. komplexními sítěmi a s jejich vlastnostmi, které se nejčastěji zkoumají.
2. Seznamte se s různými modely komplexních sítí a s možnými způsoby generování sítí.
3. Vyberte vhodné algoritmy pro generování sítí a naimplementujte je.
4. Nad vygenerovanými daty (sítěmi) proveďte experimenty (otestujte jejich vybrané vlastnosti) a výsledky experimentů vhodně reprezentujte.

### Seznam doporučené odborné literatury:

- [1] M. E. J. Newman, Networks: An Introduction, Oxford University Press (2010), ISBN-10: 0199206651.  
[2] Dále podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **RNDr. Eliška Ochodková, Ph.D.**

Datum zadání: 01.09.2015

Datum odevzdání: 28.04.2017



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2017

*Jakub Válek*  
.....

Rád bych na tomto místě poděkoval vedoucí práce RNDr. Elišce Ochodkové, Ph.D., za odborné rady, vstřícnost, poskytnuté studijní materiály a konzultace.

## **Abstrakt**

Tato bakalářská práce se zabývá generováním sítí podle několika modelů. Konkrétně se jedná o náhodný graf, zobecněný náhodný graf zahrnující konfigurační model a náhodný graf s očekávanou stupňovou distribucí, dále o modely malého světa, do kterých spadá Watts-Strogatz model a Watts-Newman model. V neposlední řadě také o model bezškálové sítě – Barabási-Albert model. Součástí teoretické části je vysvětlení základních pojmů z teorie grafů, uvedení několika zkoumaných vlastností sítí a hlavně popis jednotlivých modelů. Součástí praktické části je vytvoření aplikace generující sítě podle modelů a provedení experimentů s daty vygenerovanými z aplikace.

**Klíčová slova:** generování sítí, vlastnosti sítí, modely sítí, náhodné grafy, modely malého světa, Barabási-Albert model

## **Abstract**

This bachelor thesis deals with networks generation according to several models. Specifically, random graph, generalized random graph including configuration model and random graph with given expected degree. It also deals with models of the small world which include the Watts-Strogatz model and the Watts-Newman model. Last but not least, model of scale-free network – Barabási-Albert model. The theoretical part explains the basic concepts of graph theory, introduces several researched network properties and especially describes individual models. The goal of the practical part is to create an application for generating networks according to models and to perform experiments with data generated from the application.

**Key Words:** networks generation, network properties, network models, random graphs, models of the small world, Barabási-Albert model

# Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam tabulek	10
Seznam výpisů zdrojového kódu	11
<b>1 Úvod</b>	<b>12</b>
<b>2 Komplexní sítě a jejich vlastnosti</b>	<b>13</b>
2.1 Základy teorie grafů . . . . .	13
2.2 Zkoumané vlastnosti komplexních sítí . . . . .	17
<b>3 Modely komplexních sítí</b>	<b>22</b>
3.1 Erdős-Rényi model – náhodný graf . . . . .	22
3.2 Zobecněný náhodný graf . . . . .	26
3.3 Modely malého světa . . . . .	30
3.4 Modely bezškálových sítí – Barabási-Albert Model . . . . .	34
<b>4 Implementace aplikace pro generování sítí</b>	<b>39</b>
4.1 Použité struktury . . . . .	40
4.2 Návrh aplikace . . . . .	41
4.3 Implementace algoritmů modelů . . . . .	45
<b>5 Experimenty nad vygenerovanými daty</b>	<b>48</b>
<b>6 Závěr</b>	<b>57</b>
<b>Literatura</b>	<b>58</b>
<b>Přílohy</b>	<b>60</b>
<b>A Sekvenční diagram vytvoření a uložení grafu</b>	<b>60</b>
<b>B Příklady reálných sítí</b>	<b>61</b>
<b>C Výpisy kódu pro generování náhodných grafů</b>	<b>62</b>
<b>D Výpis kódu pro generování náhodného grafu s očekávanou stupňovou distribucí</b>	<b>63</b>

<b>E</b>	<b>Výpisy kódu pro generování grafů podle modelů malého světa</b>	<b>64</b>
<b>F</b>	<b>Power-law a CCDF</b>	<b>65</b>
<b>G</b>	<b>Přílohy na přiloženém DVD</b>	<b>66</b>

## Seznam použitých zkratek a symbolů

BA model	– Barabási-Albert model, model bezškálové sítě
BFS	– Prohledávání do šířky
$C$	– Shlukovací koeficient grafu
$\langle C \rangle$	– Průměrný shlukovací koeficient grafu
CCDF	– Doplnková kumulativní distribuční funkce
DFS	– Prohledávání do hloubky
$E$	– Množina hran grafu
ER model	– Erdős-Rényi model, model náhodného grafu
$G$	– Graf jako uspořádaná dvojice $G = (V, E)$
ID	– Identifikátor
$K_n$	– Kompletní graf na $n$ vrcholech
$\mathbb{N}$	– Množina přirozených čísel
NG_DegDist	– Náhodný graf s očekávanou stupňovou distribucí
$O(f)$	– Množina všech funkcí, které rostou nejvýše tak rychle jako $f$
$\Pr(k)$ nebo $p_k$	– Pravděpodobnost, že náhodně vybraný vrchol bude stupně $k$
$\mathbb{R}$	– Množina reálných čísel
$V$	– Množina vrcholů grafu
WN model	– Watts-Newman model, model malého světa
WS model	– Watts-Strogatz model, model malého světa
WWW	– World Wide Web
$d$	– Průměr grafu
$k_i$ nebo $\deg(i)$	– Stupeň vrcholu $i$ grafu
$\langle k \rangle$	– Průměrný stupeň vrcholů grafu
$\langle l \rangle$	– Průměrná nejkratší cesta grafu
$m$ nebo $ E(G) $	– Počet hran grafu $G$
$n$ nebo $ V(G) $	– Počet vrcholů grafu $G$
$p$	– Pravděpodobnost spojení/přepojení dvou hran
$w(e)$	– Váha hrany $e$
$\alpha$	– Exponent mocninného zákona



## Seznam obrázků

1	Příklad (jednoduchého) grafu. . . . .	14
2	Kompletní graf $K_5$ . . . . .	15
3	Všechny grafy o třech vrcholech a dvou hranách. . . . .	23
4	Výřez ze dvou grafů porovávajících binomické a Poissonovo rozdělení. . . . .	25
5	Příklad generování grafu podle konfiguračního modelu. . . . .	27
6	Změna vlastností grafu s rostoucím $p$ . . . . .	30
7	Průběh průměru a shlukovacího koeficientu WS modelu v závislosti na $p$ . . . . .	31
8	Porovnání Poissonova rozdělení a power-law. . . . .	35
9	Grafické znázornění matice sousednosti a seznamu sousedů pro jednoduchý neorientovaný graf. . . . .	40
10	Třída <code>Graph</code> . . . . .	41
11	Třídní diagram s třídami pro reprezentaci seznamu sousedů. . . . .	42
12	Třídní diagram zachycující vztah orientovaného a neorientovaného grafu. . . . .	42
13	Reprezentace hrany grafu v aplikaci. . . . .	43
14	Třída <code>GraphProperties</code> pro výpočty vlastností grafu. . . . .	44
15	Rozhraní modelu. . . . .	44
16	Vizualizace vygenerovaných grafů. . . . .	49
17	Distribuce stupňů vygenerovaných náhodných grafů s různými parametry. . . . .	50
18	Distribuce stupňů vygenerovaných zobecněných náhodných grafů s různými parametry v log-log měřítku. . . . .	51
19	Distribuce stupňů grafů vygenerovaných podle modelů malého světa. . . . .	52
20	Distribuce stupňů grafů vygenerovaných podle Barabási-Albert modelu. . . . .	53
21	Distribuce stupňů grafů BA modelu s použitím logaritického „binningu“. . . . .	54
22	Distribuce stupňů grafů BA modelu s použitím logaritického „binningu“, vydělená $2m^2$ . . . . .	54
23	Určení exponentu $\alpha$ mocninného zákona z distribuce stupňů BA modelu použitím logaritického „binningu“. . . . .	55
24	Určení exponentu $\alpha$ mocninného zákona z distribuce stupňů BA modelu použitím CCDF. . . . .	56
25	Sekvenční diagram vytvoření a uložení grafu. . . . .	60

## Seznam tabulek

1	Asymptotické časové složitosti metod třídy <i>Graph</i> v nejhorším případě. . . . .	43
2	Asymptotické časové složitosti metod třídy <i>GraphProperties</i> pro výpočty nad grafem a algoritmy, na kterých jsou tyto metody založeny. . . . .	45
3	Některé vlastnosti vygenerovaných náhodných grafů s různými parametry. . . . .	50
4	Některé vlastnosti vygenerovaných zobecněných náhodných grafu s různými parametry. . . . .	51
5	Některé vlastnosti vygenerovaných grafů podle modelů malého světa. . . . .	52
6	Některé vlastnosti vygenerovaných grafů podle Barabási-Albert modelu. . . . .	53
7	Reálné sítě a jejich základní vlastnosti. . . . .	61

## Seznam výpisů zdrojového kódu

1	Inicializace seznamů předchůdců a následovníků. . . . .	42
2	Generování konfiguračního modelu. . . . .	46
3	Iterátor kruhové mřížky. . . . .	46
4	Vytvoření počátečního grafu BA modelu. . . . .	47
5	Získání sousedů pro nový vrchol BA modelu podle preferenčního připojování. . .	47
6	Generování Barabási-Albert modelu. . . . .	47
7	Generování náhodného $G(n, p)$ grafu. . . . .	62
8	Generování náhodného $G(n, m)$ grafu. . . . .	62
9	Generování náhodného grafu s očekávanou stupňovou distribucí. . . . .	63
10	Vytvoření mřížky pro modely malého světa. . . . .	64
11	Generování WS modelu. . . . .	64
12	Generování WN modelu. . . . .	64

# 1 Úvod

Sítě se vyskytují všude kolem nás. Může se jednat o silniční síť spojující města, telefonní síť spojující lidi nebo internet tvořený spojeními mezi routery. Síť může být reprezentován i lidský mozek jako síť vzájemně komunikujících neuronů. Z pozorování takových sítí vyplývá, že ne vždy jsou jednotlivá spojení zcela náhodná. Ve skutečnosti je to spíše naopak. V reálných sítích lze pozorovat určité opakující se vzorce. Časem vznikaly různé modely snažící se napodobit reálné síť. Cílem této práce je popsat a prozkoumat několik existujících modelů, implementovat aplikaci generující síť (grafy) podle těchto modelů a prostřednictvím experimentů s vygenerovanými grafy z této aplikace porovnat vlastnosti zkoumaných modelů.

Protože síť reprezentujeme pomocí grafů, budou v úvodní kapitole zmíněny základní pojmy z teorie grafů a také některé vybrané zkoumané vlastnosti reálných sítí. Poté následuje kapitola věnující se několika modelům, které bývají v literatuře zmiňovány nejčastěji.

Druhou část práce tvoří praktická část. V ní bude nejdříve stručně popsán návrh aplikace a implementace algoritmů generující jednotlivé modely. Nakonec budou provedeny experimenty nad vygenerovanými daty. Ty se budou týkat hlavně ověření teorie o modelech z první části práce.

## 2 Komplexní síť a jejich vlastnosti

Příkladem reálných komplexních sítí mohou být například síť sociální (rodina), informační (internet), biologické (síť neuronů v mozku), dopravní (silniční síť). Další příklady reálných sítí jsou v přílohách v tabulce 7. Síť lze zobrazit prostřednictvím grafů. Pro správné porozumění a pochopení následujících kapitol je vhodné grafy definovat a popsat jejich základní vlastnosti.

### 2.1 Základy teorie grafů

Graf si můžeme představit jako množinu vrcholů. Ty lze zakreslit třeba kolečky nebo body (také čtverečky nebo trojúhelníky), které jsou vzájemně propojeny křivkami. V této kapitole jsou shrnuty pouze některé základní vlastnosti grafů, především pak vlastnosti potřebné k pochopení následujících kapitol. Formální definice grafu je následující [13].

**Definice 1** *Graf  $G$  (také jednoduchý graf) je uspořádaná dvojice  $G = (V, E)$ , kde  $V^1$  je neprázdná množina vrcholů a  $E^2$  je množina hran – množina (některých) dvouprvkových podmnožin množiny  $V$ .*

*Jednoduchý graf je graf bez smyček a multihran. Jiné typy grafu, které dovolují smyčky nebo multihrany jsou uvedeny v kapitole 2.1.1.*

*Počet vrcholů grafu  $G$  se značí  $|V(G)|$ , kde  $V(G)$  je neprázdná množina vrcholů grafu  $G$ . Dále v textu bude v souvislosti se sítěmi počet hran označován také  $n$ . Například v silničních sítích jsou vrcholy města, v sociálních sítích lidé apod.*

*Počet hran grafu  $G$  se značí  $|E(G)|$ , kde  $E(G)$  je množina (i prázdná) hran grafu  $G$ . Nadále bude počet hran označován také  $m$ . Hranou může být cesta mezi městy nebo jí může být reprezentováno přátelství dvou lidí.*

O vrcholech  $u$  a  $v$  grafu  $G$  řekneme, že to jsou *sousední vrcholy*, existuje-li mezi nimi hrana. Jinak řečeno platí, že dvouprvková množina  $\{u, v\}^3 \in E(G)$ .

Na obrázku 1 je příklad grafu  $G$ . Pro tento graf  $G$  je množina vrcholů  $V(G) = \{1, 2, 3, 4, 5, 6, 7\}$ , počet vrcholů  $|V(G)| = 7$ , množina hran  $E(G) = \{\{1, 2\}, \{1, 5\}, \{2, 7\}, \{3, 4\}, \{4, 5\}, \{4, 7\}, \{5, 6\}, \{5, 7\}\}$  a počet hran  $|E(G)| = 8$ .

*Stupeň vrcholu je důležitá vlastnost každého vrcholu. Je dán počtem hran zasahujících do vrcholu. O vztahu vrcholu a hrany, kdy hrana do daném vrcholu zasahuje, se říká, že vrchol je s touto hranou *incidentní*. Stupeň vrcholu je tedy počet hran s ním incidentních, viz definice z [13].*

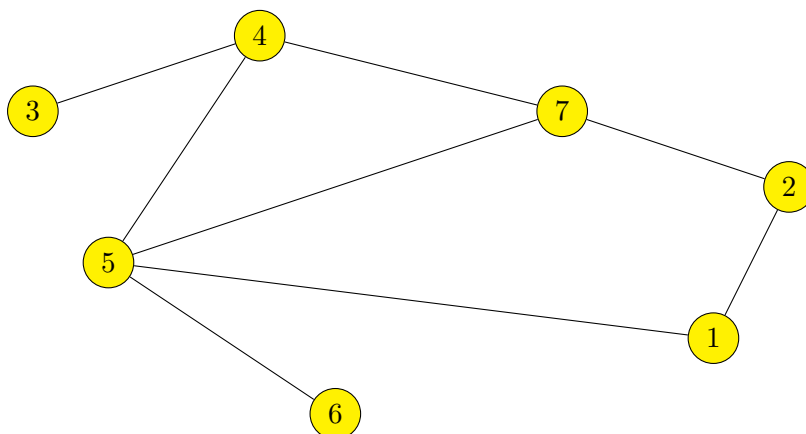
**Definice 2** *Stupeň vrcholu  $v$  grafu  $G$  je definován jako počet hran, se kterými je vrchol incidentní.*

---

<sup>1</sup>Z anglického Vertex (vrchol).

<sup>2</sup>Z anglického Edge (hrana).

<sup>3</sup>V množinách nezáleží na pořadí prvku, množiny  $\{u, v\}$  a  $\{v, u\}$  jsou shodné.



Obrázek 1: Příklad (jednoduchého) grafu.

Stupeň vrcholu  $v$  grafu  $G$  se značí  $\deg(v)$  nebo  $k$ ,  $k_i$  je pak stupeň  $i$ -tého vrcholu. Pro graf na obrázku 1 je  $k_1 = 2$ ,  $k_5 = 4$ , atd. Více o stupních i pro orientované grafy v kapitole 2.2.1. V sociálních sítích by stupeň vrcholu představoval například počet přátel konkrétního člověka.

Se stupněm vrcholu souvisí i další důležitá vlastnost, které se říká *princip sudosti*. Každá hrana v grafu je incidentní se dvěma vrcholy. Existence hrany tak zvýší stupeň obou vrcholů o jeden. Proto pak platí věta [13]:

**Věta 1** *Součet stupňů všech vrcholů v grafu je vždy sudý a je roven dvojnásobku počtu hran.*

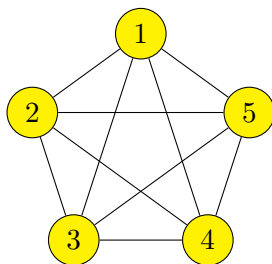
To se dá matematicky zapsat jako

$$m = \frac{1}{2} \sum_{i=1}^n k_i. \quad (1)$$

Speciálním případem jednoduchého grafu je graf, jehož každý vrchol je spojen hranou se všemi ostatními vrcholy. Tento graf se nazývá *kompletní graf*. Většinou se označuje  $K_n$ , kde  $n$  je počet vrcholů a je definován následovně [13]:

**Definice 3** *Graf na  $n$  vrcholech, kde  $n \in \mathbb{N}$  který obsahuje všech  $\binom{n}{2}$  hran se nazývá úplný nebo také kompletní graf a značí se  $K_n$ .*

Každý vrchol kompletního grafu s  $n$  vrcholy má tak  $n - 1$  sousedů. Znamená to tedy, že žádný jednoduchý graf nemůže mít více než  $\binom{n}{2}$  hran. Tohoto vztahu se často využívá při výpočtech. Příklad kompletního grafu  $K_5$  je na obrázku 2.



Obrázek 2: Kompletní graf  $K_5$ .

Hrany grafu (sítě) mohou mít také vlastnosti, které určují jeho charakter. Grafy tak lze dále dělit:

- Podle orientace hran
  - Orientované
  - Neorientované
- Podle ohodnocení hran (vrcholů)
  - Ohodnocené
  - Neohodnocené

Graf, jehož hrany jsou orientované se nazývá *orientovaný graf*. Na rozdíl od neorientované hrany, která reprezentuje obousměrný vztah vrcholů s ní incidentních, orientovaná hrana reprezentuje pouze jednosměrný vztah. Orientovaná hrana se nejčastěji zakresluje šipkou na jednom konci hrany. Orientovanou hranou lze reprezentovat například vztah rodič - dítě.

**Definice 4** *Orientovaným grafem rozumíme uspořádanou dvojici  $G = (V, E)$ , kde  $V$  je množina vrcholů a množina orientovaných hran je  $E \subseteq V \times V$ . [13]*

Z definice vyplývá, že orientovaná hrana pak není dvouprvkovou podmnožinou množiny vrcholů, ale uspořádanou dvojicí  $(v_1, v_2)$ , kde  $v_1, v_2 \in V$  a současně platí, že  $(v_1, v_2) \neq (v_2, v_1)$ .

Graf, jehož hrany jsou ohodnocené se nazývá *ohodnocený graf*. V ohodnoceném grafu se hrana zakresluje společně s číslem určujícím váhu hrany. Ohodnocené hrany lze použít například pro reprezentaci silniční sítě, kde vrcholy jsou města, hrany silnice a váha hrany je délka silnice mezi městy.

**Definice 5** *Ohodnocení grafu  $G$  je funkce  $w : E(G) \rightarrow \mathbb{R}$ , která každé hraně  $e \in E(G)$  přiřadí reálné číslo  $w(e)$ , kterému říkáme váha hrany (značení  $w$  pochází z anglického „weight“). Ohodnocený graf je graf  $G$  spolu s ohodnocením hran reálnými čísly.*

*Kladně ohodnocený (říkáme také vážený) graf  $G$  má takové ohodnocení  $w$ , že pro každou hranu  $e \in E(G)$  je její váha  $w(e)$  kladná. [13]*

Podle definice mohou být čísla přiřazená hranám (váhy hran) libovolná, tedy i záporná. Většinou se ale setkáváme s kladně ohodnocenými hranami ( $w(e) \geq 1$ ).

Pro potřeby tohoto textu budou ještě zavedeny následující definice [13].

**Definice 6** *Sledem  $v_0v_n$  v grafu  $G$  rozumíme takovou posloupnost vrcholů a hran*

$$(v_0, e_1, v_1, e_2, v_2, \dots, e_n, v_n),$$

*kde  $v_i$  jsou vrcholy grafu  $G$  a  $e_i$  jsou hrany grafu  $G$ , přičemž každá hrana  $e_i$  má koncové vrcholy  $v_{i-1}$  a  $v_i$ . Počet hran nazveme délkou sledu  $v_0v_n$ . Vrchol  $v_0$  nazýváme počáteční a vrchol  $v_n$  koncový.*

**Definice 7** *Tah je sled, ve kterém se neopakují žádné hrany a cesta je sled, ve kterém se neopakují žádné vrcholy (tedy ani hrany).*

Protože některé modely v následujícím textu generují nesouvislé grafy, je vhodné zadefinovat pojem „souvislost“ [13].

**Definice 8** *Řekneme, že vrchol  $v$  je dosažitelný z vrcholu  $u$ , jestliže v grafu existuje sled z vrcholu  $u$  do vrcholu  $v$ .*

*Graf nazveme souvislý, jestliže pro každé dva vrcholy  $u, v$  je vrchol  $v$  dosažitelný z vrcholu  $u$ . V opačném případě je graf nesouvislý.*

Nesouvislý graf se pak skládá z několika „částí“, které se nazývají „komponenty“ [13]. V orientovaných grafech rozlišujeme, zda je graf silně nebo slabě souvislý. Orientovaný graf je *slabě souvislý*, jestliže pro každé dva vrcholy  $u$  a  $v$  existuje neorientovaná cesta z  $u$  do  $v$  (tedy orientovaná cesta z  $u$  do  $v$  nebo z  $v$  do  $u$ ). Dále je orientovaný graf *silně souvislý*, jestliže pro každé dva vrcholy  $u$  a  $v$  existuje orientovaná cesta z  $u$  do  $v$  a orientovaná cesta z  $v$  do  $u$  [10]. Obdobně hovoříme o silně a slabě souvislých komponentách orientovaných grafů.

### 2.1.1 Další typy grafů

Nejčastěji se setkáváme a pracujeme s jednoduchým grafem podle definice 1. Existují ale také grafy, které na rozdíl od jednoduchého grafu dovolují vznik smyček nebo násobných hran (multihran).

**Smyčka** je hrana, která končí ve stejném vrcholu, ve kterém začíná.

**Násobné hrany** nebo také *multihrany* jsou hrany, které spojují stejné vrcholy. Mezi dvěma vrcholy tedy existuje více hran.

Graf, ve kterém se vyskytují násobné hrany (a mohou, ale nemusí i smyčky), se nazývá *multigraf*.



## 2.2 Zkoumané vlastnosti komplexních sítí

V této kapitole budou zmíněny některé zkoumané vlastnosti komplexních sítí. Za nejčastěji zkoumané vlastnosti lze považovat *distribuci stupňů*, *průměrný stupeň*, *shlukovací koeficient*, *průměr sítě* a další jako počet hran, počet vrcholů apod. Tyto charakteristiky také umožňují porovnávat jednotlivé modely komplexních sítí mezi sebou. Samozřejmě existuje i spousta dalších vlastností, kterými se lze zabývat, například *počet komponent*, *hustota* apod.

### 2.2.1 Průměrný stupeň a distribuce stupňů

Důležitou vlastností sítí je *průměrný stupeň* vypočtený jako aritmetický průměr stupňů všech vrcholů v grafu. Pro neorientovaný graf lze průměrný stupeň vyjádřit následovně

$$\langle k \rangle = \frac{1}{n} \sum_{i=1}^n k_i. \quad (2)$$

Využitím věty 1 (princip sudosti, strana 14) lze vztah přepsat na

$$\langle k \rangle = \frac{2m}{n}. \quad (3)$$

V orientovaných sítích se rozlišuje *vstupní stupeň*, *výstupní stupeň* a *celkový stupeň*. Vstupní stupeň, označován  $k_i^{in}$ , reprezentuje počet hran vedoucích do  $i$ -tého vrcholu. Výstupní stupeň,  $k_i^{out}$ , reprezentuje počet hran vycházejících z  $i$ -tého vrcholu a celkový stupeň  $k_i$   $i$ -tého vrcholu je pak [1]

$$k_i = k_i^{in} + k_i^{out}. \quad (4)$$

V orientovaných grafech také platí  $\sum_{i=1}^n k_i^{in} = \sum_{i=1}^n k_i^{out}$ , a to proto, že každá orientovaná hrana musí začínat i končit v nějakém vrcholu. Každá hrana tak přidává po jednom stupni do sumy vstupních stupňů a do sumy výstupních stupňů. Tato rovnost tedy také odpovídá počtu hran v orientovaných grafech

$$m = \sum_{i=1}^n k_i^{in} = \sum_{i=1}^n k_i^{out}. \quad (5)$$

Při výpočtu průměrného stupně v orientovaných sítích je pak jedno, zda počítáme se vstupními nebo výstupními stupni

$$\langle k^{in} \rangle = \langle k^{out} \rangle = \frac{1}{n} \sum_{i=1}^n k_i^{in} = \frac{1}{n} \sum_{i=1}^n k_i^{out} = \frac{m}{n}. \quad (6)$$

*Distribuce stupňů* je další důležitá vlastnost sítí. Značí se  $p_k$ , popřípadě  $\Pr(k)$  a interpretuje se jako pravděpodobnost, že náhodně vybraný vrchol grafu bude stupně  $k$  (bude mít  $k$  sousedů), kde  $p_k \in (0, 1)$ . Pro síť na  $n$  vrcholech lze  $p_k$  vypočítat jako

$$p_k = \frac{n_k}{n}, \quad (7)$$

kde  $n_k$  je počet vrcholů se stupněm  $k$ . Protože distribuce stupňů je rozdělením pravděpodobnosti, platí také<sup>4</sup>

$$\sum_{k=0}^{\infty} p_k = 1. \quad (8)$$

Pokud nechceme počítat s nekonečnem, lze pro jednoduchý graf vztah zjednodušit. Protože v jednoduchém grafu má každý vrchol maximálně  $n - 1$  sousedů, nemůže existovat vrchol, který by měl stupeň vyšší než právě  $n - 1$ , a potom  $\sum_{k=0}^{n-1} p_k = 1$ .

### 2.2.2 Shlukovací koeficient

Shlukovací koeficient, označován  $C$ , vyjadřuje tendenci vrcholů tvořit shluky. Máme graf s vrcholy A, B, C, kde vrchol A je sousedem vrcholu B a vrchol B sousedem vrcholu C. Shlukovacím koeficientem je vyjádřena pravděpodobnost, že i vrchol A bude sousedem vrcholu C. Jinými slovy mezi vrcholy A, B, C existuje tranzitivní relace. Proto se někdy místo termínu shlukování používá termín tranzitivita. Můžeme se setkat se dvěma definicemi shlukovacího koeficientu: *lokální shlukovací koeficient* a *globální shlukovací koeficient*.

*Globální shlukovací koeficient* popisuje graf jako celek a je definován jako [15]

$$C = \frac{3 \times \text{počet trojúhelníků v grafu}}{\text{počet spojených trojic vrcholů}}, \quad (9)$$

kde „spojená trojice vrcholů“ je chápána jako vrchol se dvěma hranami vedoucími do dvojice vrcholů. Stojí za povšimnutí, že každý trojúhelník (podgraf  $K_3$ ) má právě tři spojené trojice vrcholů. Proto hodnota shlukovacího koeficientu  $C$  bude v intervalu  $C \in \langle 0, 1 \rangle$ . Shlukovací koeficient tak vyjadřuje pravděpodobnost, že dva sousedé nějakého vrcholu jsou navzájem také sousedy.

*Lokální shlukovací koeficient* popisuje tendenci tvořit shluky okolo konkrétního vrcholu v grafu. Může být tedy vypočten pro každý vrchol následovně [15]:

$$C_i = \frac{\text{počet trojúhelníků v grafu obsahujících vrchol } i}{\text{počet spojených trojic se středem ve vrcholu } i}, \quad (10)$$

kde „spojená trojice se středem ve vrcholu  $i$ “ je chápána obdobně jako ve vzorci 9. Pro vrcholy, jejichž stupeň je 0 nebo 1 pokládáme  $C_i = 0$  (vyhneme se tak dělení nulou). Poté můžeme vypočíst z lokálního shlukovacího koeficientu také shlukovací koeficient pro celý graf (jak činí globální shlukovací koeficient) jako aritmetický průměr

$$\langle C \rangle = \frac{1}{n} \sum_{i=1}^n C_i, \quad (11)$$

---

<sup>4</sup>Pokud distribuci stupňů považujeme za rozdělení pravděpodobnosti spojitě náhodné veličiny, pak platí  $\int_{k_{min}}^{\infty} p_k dk = 1$ .

ovšem výsledky se budou od těch získaných ze vzorce 9 mírně lišit. Proto je dobré rozlišovat použitou metodu. Pro obě metody však platí, že pro reálné „lidské“ sítě bude hodnota shlukovacího koeficientu vyšší, než by tomu bylo v případě náhodného grafu.

Pro neorientovaný neohodnocený graf lze vzorec 10 pro výpočet lokálního shlukovacího koeficientu přepsat na [1]

$$C_i = \frac{L_i}{\binom{k_i}{2}} = \frac{2L_i}{k_i(k_i - 1)}, \quad (12)$$

kde  $L_i$  je počet spojení mezi sousedy  $i$ -tého vrcholu a  $\binom{k_i}{2}$  je maximální počet těchto spojení mezi sousedy. U tohoto vzorce si stačí jen představit vrchol  $v$  stupně  $k$  ( $k$  sousedů). Pak každá hrana mezi těmito sousedy vytvoří trojúhelník obsahující vrchol  $v$  a „maximální počet těchto spojení mezi sousedy“ reprezentuje „spojenou trojici se středem ve vrcholu  $v$ “, protože potřebné dvě hrany jsou už zajištěny z vrcholu  $v$ .

Shlukovací koeficient je definován také pro ohodnocené grafy a lze jej poměrně jednoduše vypočítat pro graf reprezentovaný maticí sousednosti, viz [12]. Pro ohodnocené grafy existuje více definic shlukovacího koeficientu, jedna z nich je následující [11, 17]:

$$\tilde{C}_i = \frac{\sum_{j,k} (\hat{w}_{ij}\hat{w}_{ik}\hat{w}_{jk})^{\frac{1}{3}}}{k_i(k_i - 1)}, \quad (13)$$

kde váhy hran náleží do intervalu  $\langle 0, 1 \rangle$  a pokud tomu tak není, musí být normalizovány maximální vahou hrany v grafu, tedy  $\hat{w}_{ij} = \frac{w_{ij}}{\max(w)}$ .

### 2.2.3 Průměr a průměrná nejkratší cesta

*Nejkratší cesta* značená  $l_{ij}$  je nejkratší cesta mezi dvěma vrcholy  $v_i$  a  $v_j$  grafu  $G$ , tedy cesta<sup>5</sup> z vrcholu  $v_i$  do vrcholu  $v_j$  přes co nejmenší počet hran.

V sítích se pak zkoumá průměrná nejkratší cesta a průměr sítě. *Průměrná nejkratší cesta* je aritmetický průměr nejkratších cest mezi všemi možnými dvojicemi vrcholů v grafu. Pro neorientovaný graf ji lze vypočítat jako [20]

$$\langle l \rangle = \frac{1}{\binom{n}{2}} \sum_i^n \sum_{j>i}^n l_{ij} = \frac{2}{n(n-1)} \sum_i^n \sum_{j>i}^n l_{ij}, \quad (14)$$

pro orientovaný graf pak obdobně [20]

$$\langle l \rangle = \frac{1}{n(n-1)} \sum_i^n \sum_j^n l_{ij}. \quad (15)$$

Vzorec 15 lze použít i pro neorientovaný graf, protože vzorec 14 pouze využívá faktu, že matice sousednosti neorientovaného grafu je osově symetrická podle diagonály, a tak stačí projít jen

---

<sup>5</sup>Z definice 7 na straně 16.

její polovinu. Pokud by síť obsahovala více komponent, existovala by alespoň jedna dvojice vrcholů  $v_i, v_j$  mezi nimiž by nejkratší cesta  $l_{ij} = \infty$ . Průměrná nejkratší cesta by pak také divergovala, a proto se v sítích obsahujících více komponent počítá průměrná nejkratší cesta největší komponenty.

*Průměrem sítě  $d$*  se rozumí nejdelší nejkratší cesta. Jinými slovy nejdelší cesta ze všech možných nejkratších cest mezi všemi možnými dvojicemi vrcholů.

Pro výpočet průměrné nejkratší cesty i průměru konkrétního grafu lze použít algoritmus prohledávání do šířky (BFS)<sup>6</sup> pro neohodnocený graf [6, 15] nebo například Floydův algoritmus<sup>7</sup> pro ohodnocený graf [13].

#### 2.2.4 Vlastnosti reálných sítí

Protože většinou jsou zkoumané sítě sítěmi reálnými, není od věci zmínit některé očekávané vlastnosti reálných sítí. Jednak budeme vědět, co od reálných sítí očekávat, a také nám to může pomoci při rozhodování o technikách použitých při implementaci některých algoritmů. Následující informace jsou platné pro spoustu reálných sítí, přesto zde existují i výjimky [4].

*Distribuce stupňů v reálných „lidských“ sítích se často řídí mocninným zákonem (anglicky power-law).* Takovou síť představují například sociální sítě, WWW apod. Jak bylo řečeno, existují výjimky a jedním příkladem sítě, jejíž distribuce stupňů se neřídí mocninným zákonem, je silniční síť [4]. Hustotu pravděpodobnosti spojitě náhodné veličiny podle mocninného zákona lze zapsat vztahem

$$p(k) \sim k^{-\alpha}, \quad (16)$$

kde  $\alpha$  většinou nabývá hodnot v rozsahu  $2 \leq \alpha \leq 3$ . Pokud je  $\alpha < 2$ , pak by se v grafu vyskytovaly vrcholy, jejichž stupeň  $k$  by byl větší, než počet vrcholů  $n$  v grafu. Pokud je  $\alpha > 3$ , pak se objevuje „škála“ a graf se začíná podobat náhodnému grafu. Celá problematika je poměrně rozsáhlá a je popsána v [4, 9]. Rozdělení pravděpodobnosti podle power-law můžeme považovat za rozdělení jak spojitě tak i diskrétní náhodné veličiny. Pokud jej budeme považovat za rozdělení pravděpodobnosti spojitě náhodné veličiny, můžeme se setkat se vztahem

$$p(k) = \frac{\alpha - 1}{k_{min}} \left( \frac{k}{k_{min}} \right)^{-\alpha}. \quad (17)$$

Mocninný zákon s vhodně zvoleným parametrem  $\alpha$  odpovídá známému pravidlu „80:20“. V teorii sítí to znamená, že 20% vrcholů má součet stupňů rovný 80% součtu stupňů všech vrcholů.

*Průměr reálných sítí je malý.* Vlastnost vychází z Milgramova experimentu provedeném v 60. letech minulého století [15]. Účastníkům experimentu byly rozdány dopisy adresované konkrétní osobě. Aby se dopis postupně dostal k cíli, směli jej účastníci předat jen svému známému. Většina dopisů byla ztracena, nicméně část jich dorazila do cíle. Překvapivě dopisy, které došly do cíle,

<sup>6</sup>Pro každý vrchol je třeba zjistit cestu do všech ostatních vrcholů s asymptotickou složitostí  $O(n(n+m))$ .

<sup>7</sup>S asymptotickou složitostí  $O(n^3)$ .

prošly rukama v průměru pouze šesti lidí. Těto vlastnosti se říká také „small world effect“. Uvádí se, že graf má „small world“ vlastnost, pokud průměrná nejkratší cesta  $\langle l \rangle$  roste logaritmicky (nebo pomaleji) s počtem vrcholů  $n$ , tedy [2]

$$\langle l \rangle \sim \log n. \quad (18)$$

*Shlukovací koeficient reálných sítí je vysoký.* V reálných sítích je zvýšená pravděpodobnost, že dva vrcholy budou spojeny hranou, pokud mají společného souseda. Obdobně jako u distribuce stupňů zde lze pozorovat rozdělení podle power-law

$$C(k) \sim k^{-\alpha}, \quad (19)$$

kde  $C(k)$  je průměrný shlukovací koeficient všech vrcholů stupně  $k$  [20].

*Hustota reálných sítí je malá.* Říkáme také, že jsou řídké. Jinými slovy reálné sítě často mají mnohem méně hran, než by měl kompletní graf o stejném počtu vrcholů. Obvykle se uvádí počet hran [20]

$$m = O(n). \quad (20)$$

### 3 Modely komplexních sítí

Jedním z cílů vědy zabývající se sítěmi (network science) je vytvořit modely, podle kterých by bylo možné napodobit reálné sítě (přesněji řečeno napodobit tak, aby si co nejvíce odpovídaly ve zkoumaných vlastnostech). Hlavním problémem tedy není nějakou síť vytvořit (což je poměrně jednoduché), ale zajistit, že vrcholy budou pospojovány hranami tak, aby byly splněny určité vlastnosti sítě.

V této kapitole budou probrány nejpoužívanější (nejobecnější) modely sítí vždy se základním popisem, pseudokódem pro vygenerování takové sítě a s očekávaným chováním některých vlastností popsaných v kapitole 2.2.

#### 3.1 Erdős-Rényi model – náhodný graf

Reálné sítě se na první pohled mohou jevit jako zcela náhodné. Tomu odpovídá první z modelů – Erdős-Rényi model. Tento model je pojmenován po dvou Maďarských matematicích Paulu Erdős a Alfrédu Rényi, kteří jej poprvé definovali v roce 1959 [2]. Existují dvě ekvivalentní definice náhodného grafu:

- $G(n, m)$  model, kde  $n$  je počet vrcholů a  $m$  je počet náhodně umístěných hran.
- $G(n, p)$  model, kde  $n$  je počet vrcholů a každá dvojice vrcholů je spojena s pravděpodobností  $p$ .

Erdős-Rényi model (dále označovaný také jako ER model) generuje náhodný graf tak, že náhodně „vybírá“ graf z množiny všech grafů s konstantním počtem vrcholů a hran ( $n$  a  $m$ ). Pravděpodobnost výběru libovolného grafu  $g \in G$  je pak

$$\Pr(g) = \frac{1}{|G|}, \quad (21)$$

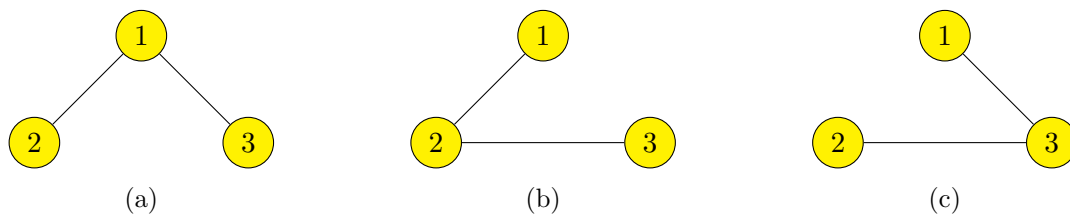
kde  $G$  je množina všech grafů s konstantním  $n$  a  $m$ . Jako příklad jsou na obrázku 3 všechny možné grafy o třech vrcholech a dvou hranách. ER model náhodně vybere jeden z těchto grafů. Pravděpodobnost výběru libovolného z nich je  $\frac{1}{3}$ .

Pokud maximální možný počet hran na  $n$  vrcholech označíme jako  $M$ , kde

$$M = \binom{n}{2} = \frac{n(n-1)}{2}, \quad (22)$$

můžeme pravděpodobnost výběru libovolného grafu  $g$  s  $n$  vrcholy a  $m$  hranami přepsat jako:

$$\Pr(g) = \frac{1}{|G|} = \frac{1}{\binom{M}{m}} = \frac{1}{\binom{\frac{n(n-1)}{2}}{m}}. \quad (23)$$



Obrázek 3: Všechny grafy o třech vrcholech a dvou hranách.

Příklad generování náhodného jednoduchého grafu  $G = (V, E)$  podle  $G(n, m)$  modelu je popsán algoritmem 1.

---

**Algoritmus 1** Generování náhodného grafu podle  $G(n, m)$  modelu.

---

**Require:**  $G, m$   
**Ensure:**  $0 \leq m \leq \binom{|V|}{2}$   
**Ensure:**  $E(G) = \emptyset$   
**while**  $|E(G)| \neq m$  **do**  
     $v_1 \leftarrow \text{RANDOM}(v \in V(G))$   
     $v_2 \leftarrow \text{RANDOM}(v \in V(G))$   
    **if**  $v_1 \neq v_2$  **then**  
         $E(G) \leftarrow E(G) \cup \{\{v_1, v_2\}\}$   
    **end if**  
**end while**

---

Proces generování podle  $G(n, p)$  modelu je obdobný a může být popsán algoritmem 2.

---

**Algoritmus 2** Generování náhodného grafu podle  $G(n, p)$  modelu.

---

**Require:**  $G, p$   
**Ensure:**  $0 \leq p \leq 1$   
**Ensure:**  $E(G) = \emptyset$   
**for all**  $v_i, v_j \in V(G); i < j$  **do**  
     $x \leftarrow \text{RANDOM}(x \in \mathbb{R}; 0 \leq x < 1)$   
    **if**  $p > x$  **then**  
         $E(G) \leftarrow E(G) \cup \{\{v_i, v_j\}\}$   
    **end if**  
**end for**

---

### 3.1.1 Průměrný stupeň

Pro  $G(n, m)$  model, kde  $n$  je počet vrcholů a  $m$  je počet hran, je určit průměrný stupeň o něco jednodušší. Z principu sudosti<sup>8</sup> platí, že součet stupňů všech vrcholů je roven  $2m$ . Pak průměrný stupeň je:

$$\langle k \rangle = \frac{2m}{n}. \quad (24)$$

---

<sup>8</sup>Viz věta 1 (princip sudosti) na straně 14.

U  $G(n, p)$  modelu platí, že existence každé hrany je nezávislá na jiných hranách a pravděpodobnost této existence je rovna  $p$ . Pravděpodobnost grafu s právě  $n$  vrcholy a  $m$  hranami je dána binomickým rozdělením

$$\Pr(m) = \binom{M}{m} p^m (1-p)^{M-m}, \quad (25)$$

kde  $M$  je maximální počet hran na  $n$  vrcholech,  $\binom{M}{m}$  je počet všech grafů s  $n$  vrcholy a  $m$  hranami,  $p^m$  je pravděpodobnost existence právě  $m$  hran a  $(1-p)^{M-m}$  pravděpodobnost neexistence zbytku hran. Střední hodnota počtu hran (z vlastností binomického rozdělení) je pak  $\langle m \rangle = Mp$ . Dosazením do vztahu 24 pro průměrný stupeň  $G(n, m)$  modelu dostaneme vztah pro výpočet průměrného stupně  $G(n, p)$  modelu

$$\langle k \rangle = \frac{2Mp}{n} = \frac{2\binom{n}{2}p}{n} = (n-1)p. \quad (26)$$

### 3.1.2 Distribuce stupňů

Distribuce stupňů v náhodném grafu se řídí binomickým rozdělením [4]. Odvození je poměrně intuitivní. Pravděpodobnost, že náhodný vrchol bude mít  $k$  vazeb na sousedy je  $p^k$ , a pravděpodobnost, že nebude mít vazbu na zbytek z  $n-1$  možných sousedů je  $(1-p)^{(n-1)-k}$ . Nakonec počet možností, jak vybrat  $k$  sousedů náhodného vrcholu, je  $\binom{n-1}{k}$ . Konečný vztah je tak

$$\Pr(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}. \quad (27)$$

Většinou však zkoumáme řídké reálné sítě s velkým počtem vrcholů, a proto lze tam, kde  $n \rightarrow \infty$  a  $p$  je velmi malé, binomické rozdělení dobře aproximovat Poissonovým rozdělením [7]:

$$\Pr(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \simeq \frac{\langle k \rangle^k}{k!} e^{-\langle k \rangle}. \quad (28)$$

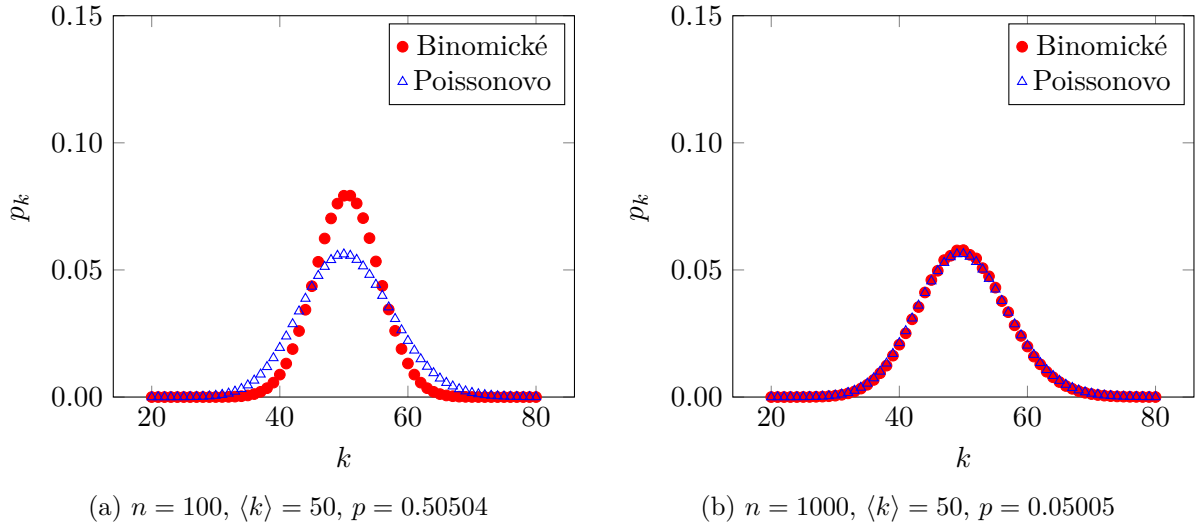
Proto se tomuto modelu někdy také říká „Poisson random graph“. Tato aproximace umožňuje zjednodušení výpočtů některých charakteristik, a proto se u náhodných grafů častěji uvádí právě Poissonovo rozdělení. Pro zájemce je odvození uvedeno v [2, 7, 20]. Porovnání binomického a Poissonova rozdělení pro různá  $n$  je na obrázku 4. Distribuce stupňů v reálných sítích se ale většinou neřídí Poissonovým rozdělením, proto ER model není příliš vhodný pro jejich popis<sup>9</sup>.

### 3.1.3 Shlukovací koeficient

Globální shlukovací koeficient v náhodných sítích je poměrně jednoduchý na výpočet. Pravděpodobnost, že dva libovolné vrcholy jsou spojeny hranou je rovna  $p$  nezávisle na jiných spojeních.

<sup>9</sup>Jedním z důvodů je neexistence takzvaných „center“ v náhodném grafu. Více v kapitole 3.4.





Obrázek 4: Výřez ze dvou grafů porovnávajících binomické a Poissonovo rozdělení. Z grafů je patrné, že už pro hodnoty  $N \geq 1000$  lze binomické rozdělení poměrně přesně aproximovat Poissonovým rozdělením [2].

Dále počet trojúhelníků v náhodném grafu je roven  $\binom{n}{3}p^3$  a počet spojených trojic je roven  $\binom{3}{2}\binom{n}{3}p^2$ . Dosazením do rovnice 9 pro výpočet globálního shlukovacího koeficientu (strana 18) dostaneme [7]

$$C = \frac{3 \times \text{počet trojúhelníků v grafu}}{\text{počet spojených trojic vrcholů}} = \frac{\binom{n}{3}p^3}{\binom{n}{3}p^2} = p = \frac{\langle k \rangle}{n-1}. \quad (29)$$

V případě lokálního shlukovacího koeficientu je odvození obdobné [20]. Vyjdeme z rovnice 10 pro výpočet lokálního shlukovacího koeficientu (strana 18) a využijeme faktu, že počet spojení  $L_i$  mezi sousedy  $i$ -tého vrcholu je  $L_i = p\binom{k_i}{2}$ . Dosazením do rovnice 10 dojdeme ke stejnému výsledku jako v předchozím případě

$$C_i = \frac{L_i}{\binom{k_i}{2}} = \frac{2L_i}{k_i(k_i-1)} = p. \quad (30)$$

Průměrná hodnota  $\langle C \rangle$  bude pak taktéž rovna  $\langle C \rangle = p$ . V případě náhodných rozsáhlých řídkých sítí (s konstantním  $\langle k \rangle$  a  $n \rightarrow \infty$ ) pak platí, že  $C \sim \frac{1}{n}$ . Pro rozsáhlé řídké sítě tak shlukovací koeficient klesá úměrně s počtem vrcholů, což příliš neodpovídá reálným sítím [2].

### 3.1.4 Průměrná nejkratší cesta

Distribuce stupňů náhodného grafu se řídí Poissonovým rozdělením a jeho střední hodnota tak bude odpovídat průměrnému stupni  $\langle k \rangle$ . To znamená, že vybereme-li náhodný vrchol  $v$ , tak ve vzdálenosti  $l = 1$  od tohoto vrcholu dosáhneme v průměru na  $\langle k \rangle$  vrcholů, ve vzdálenosti  $l = 2$  pak v průměru na  $\langle k \rangle^2$  vrcholů, atd. Počet vrcholů ve vzdálenosti  $l$  lze pak vypočítat jako součet

prvních  $l + 1$  členů geometrické posloupnosti

$$n(l) = 1 + \langle k \rangle + \langle k \rangle^2 + \dots + \langle k \rangle^l = \frac{\langle k \rangle^{l+1} - 1}{\langle k \rangle - 1} \simeq \langle k \rangle^l. \quad (31)$$

Logaritmováním a úpravou dostáváme vztah pro výpočet průměrné nejkratší cesty  $\langle l \rangle$  náhodného grafu [2, 15, 20]

$$\langle l \rangle \simeq \frac{\log n}{\log \langle k \rangle}. \quad (32)$$

Průměrná nejkratší cesta náhodného grafu tak může odpovídat malé průměrné nejkratší cestě reálných sítí (viz kapitola 2.2.4).

### 3.2 Zobecněný náhodný graf

Náhodný graf popsáný v kapitole 3.1 můžeme zobecnit a generovat graf, jehož distribuce stupňů se nemusí řídit Poissonovým rozdělením. V kapitole 3.1 byl náhodný graf definován dvěma způsoby, a sice jako  $G(n, p)$  a  $G(n, m)$  model. Oba modely lze upravit pro definování zobecněného náhodného grafu. Prvním modelem je takzvaný „*konfigurační model*“, který je analogií k  $G(n, m)$  modelu [14]. Bývá nazýván také jako „*náhodný graf s očekávanou stupňovou posloupností*“. Druhým modelem, který je analogií k  $G(n, p)$  grafu, je „*náhodný graf s očekávanou stupňovou distribucí*“.

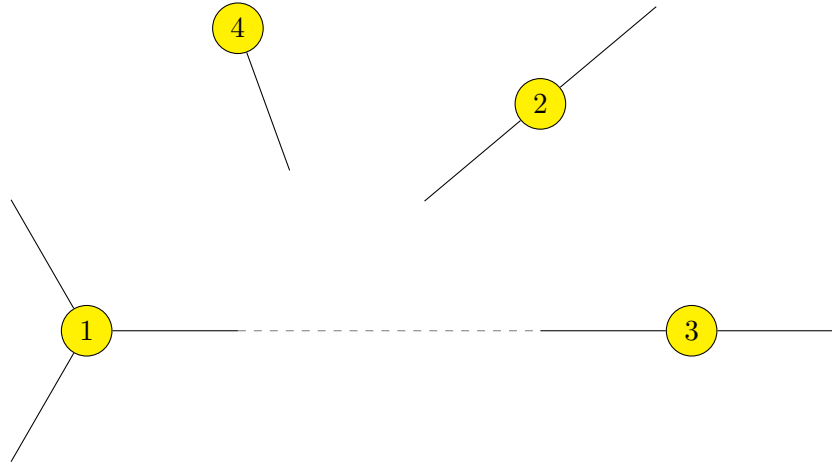
*Konfigurační model* závisí na počtu vrcholů  $n$  a posloupnosti stupňů vrcholů. Posloupnost stupňů vrcholů je množina  $n$  hodnot, kde každá hodnota  $k_i$  přesně odpovídá stupni  $i$ -tého vrcholu. Pro naše účely můžeme tuto množinu značit jako  $K$ , pro niž platí  $K = \{k_1, k_2, k_3, \dots, k_n\}$ . Pokud si vzpomeneme na princip sudosti (Věta 1), kde se také hovoří o posloupnosti stupňů vrcholů, můžeme počet hran  $m$  výsledného grafu získat jako  $m = \frac{1}{2} \sum_{i=1}^n k_i$ . Odtud vychází podobnost k  $G(n, m)$  modelu. Z principu sudosti také vyplývá podmínka pro vstupní posloupnost stupňů vrcholů, a sice že součet všech prvků posloupnosti (tedy stupňů vrcholů) musí být sudý. V opačném případě nebude možné graf sestavit. Pro konfigurační model je tedy možné, za splnění této podmínky, zvolit libovolnou posloupnost stupňů vrcholu a generovat například  $k$ -regulární grafy, kde každý vrchol má stupeň  $k$ , nebo můžeme vygenerovat posloupnost mající exponenciální, logaritmické či jiné rozdělení, popřípadě i posloupnost splňující power-law<sup>10</sup>. Dokonce můžeme použít i posloupnost stupňů vrcholů nějaké existující reálné sítě získanou pozorováním. *Distribuce stupňů  $p_k$*  je tedy plně závislá na zvolené stupňové posloupnosti a *průměrný stupeň  $\langle k \rangle$*  lze jednoduše vypočítat jako aritmetický průměr hodnot této posloupnosti.

Když známe počet vrcholů  $n$  a posloupnost stupňů vrcholů  $K$ , můžeme graf vygenerovat následujícím způsobem. Každému vrcholu  $i$  přiřadíme  $k_i$  „pahýlů“<sup>11</sup>, které představují budoucí hranu, jak je naznačeno na obrázku 5. Aby vznikla hrana, musíme spojit dva pahýly. Dostaneme tak graf s  $2m$  pahýly, kde  $m$  je počet hran. Poté náhodně vybereme dva pahýly, které spojíme.

<sup>10</sup>Viz kapitola 2.2.4.

<sup>11</sup>Anglicky označováno jako „*stubs*“.

Pokračujeme výběrem další dvojice z  $2m - 2$  pahýlů dokud nevyčerpáme všechny možnosti. Ve výsledném grafu tak bude každý vrchol mít požadovaný stupeň. Výsledný graf s konkrétním spojením pahýlů je náhodně vybrán z množiny grafů, kterou získáme permutací všech možných spojení pahýlů. Velikost této množiny je  $\binom{2m}{2}$  a pravděpodobnost každého možného spojení pahýlů pak  $\frac{1}{\binom{2m}{2}}$ . Algoritmem 3 je naznačen jeden z možných způsobů generování.



Obrázek 5: Příklad generování grafu podle konfiguračního modelu. Počet vrcholů  $n = 4$  a stupňová posloupnost  $K = \{3, 2, 2, 1\}$ . Spojením dvou náhodných pahýlů vznikne hrana (vyznačeno čárkovanou čarou).

---

**Algoritmus 3** Generování konfiguračního modelu.

---

**Require:**  $G, K$

**Ensure:**  $(\sum k_i \in K) \bmod 2 = 0$

**Ensure:**  $E(G) = \emptyset$

$J[]$  = pole, kde každý vrchol  $i \in \langle 0, |K| \rangle$  je v poli právě  $k_i$  krát.

$J[] \leftarrow PERMUTATION(J[])$

▷ Náhodná permutace prvků v poli

$i \leftarrow 0$

**while**  $i < |J[]$  **do**

$E(G) \leftarrow E(G) \cup ((v_i, v_{i+1}))$

$i \leftarrow i + 2$

**end while**

---

Konfigurační model implicitně umožňuje vznik smyček a multihran, ovšem s  $n \rightarrow \infty$  je pravděpodobnost jejich vzniku nulová [14]. Lze si poměrně snadno představit takovou modifikaci postupu generování, která by jejich vznik znemožnila. Avšak taková modifikace není doporučena [14], protože pak by každé možné spojení pahýlu nebylo stejně pravděpodobné, což by znemožnilo analyticky vypočítat některé vlastnosti modelu. Nicméně fakt, že každé možné spojení pahýlů je stejně pravděpodobné automaticky neznamená, že také každý výsledný graf je stejně pravděpodobný (některá spojení generují stejné grafy). Podrobně je tato problematika popsána v [14], kde je ukázáno, že pro  $n \rightarrow \infty$  je opravdu každý možný výsledný graf stejně pravděpodobný.

Jak bylo zmíněno, druhým modelem je *náhodný graf s očekávanou stupňovou distribucí*, který je analogií k  $G(n, p)$  modelu z kapitoly 3.1. Vstupem je opět počet vrcholů  $n$  a posloupnost hodnot  $K = \{k_1, k_2, \dots, k_n\}$ . Na rozdíl od konfiguračního modelu  $k_i$  nepředstavuje „fixní“ stupeň vrcholu  $i$ , ale očekávaný stupeň vrcholu  $i$ .

Pro model je podstatná pravděpodobnost  $p_{ij}$  existence hrany mezi libovolnými vrcholy  $i$  a  $j$ , jejíž výpočet vychází z konfiguračního modelu. Z posloupnosti známe  $k_i$  a  $k_j$ . Vyberme jeden konkrétní pahýl vrcholu  $i$ . Tento pahýl se připojí k jednomu z  $k_j$  pahýlů vrcholu  $j$ . Celkem je v grafu  $2m$  pahýlů pro připojení, z toho  $2m - 1$  použitelných (nemůže se napojit sám na sebe). Pravděpodobnost  $\frac{k_j}{2m-1}$  připojení konkrétního pahýlu vrcholu  $i$  k nějakému pahýlu vrcholu  $j$  stačí vynásobit počtem pahýlů  $k_i$  vrcholu  $i$ , čímž dostaneme vztah pro výpočet pravděpodobnosti existence hrany mezi vrcholy  $i$  a  $j$

$$p_{ij} = \frac{k_i k_j}{2m - 1}. \quad (33)$$

Pro situace, kdy graf má velký počet hran  $m$ , můžeme  $-1$  ve jmenovateli ignorovat a psát

$$p_{ij} = \frac{k_i k_j}{2m}. \quad (34)$$

Nesmíme ale zapomenout na smyčky, kde počet párů v čitateli nebude  $k_i k_i$ , ale  $\binom{k_i}{2} = \frac{k_i(k_i-1)}{2}$ . Proto pravděpodobnost smyčky ve vrcholu  $i$  je

$$p_{ii} = \frac{k_i(k_i - 1)}{4m}. \quad (35)$$

Při generování grafu tak můžeme pro každou dvojici vrcholů postupovat podle vztahu<sup>12</sup> uvedeného v [14]

$$p_{ij} = \begin{cases} \frac{k_i k_j}{2m} & \text{pokud } i \neq j, \\ \frac{k_i^2}{4m} & \text{pokud } i = j. \end{cases} \quad (36)$$

Vygenerování konkrétního náhodného grafu s očekávanou stupňovou distribucí může být dosaženo například algoritmem 4.

### 3.2.1 „Excess degree distribution“

Mnoho vlastností konfiguračního modelu včetně shlukovacího koeficientu lze vypočítat pomocí tzv. *excess degree distribution* [8, 14]. *Excess degree distribution* je rozdělení pravděpodobnosti stupňů náhodného souseda náhodně vybraného vrcholu bez započtení hrany, kterou byl tento soused dosažen.

Toto krkolonné vysvětlení bude možná snáze pochopitelné z odvození. Začneme zjednodušením problému. Vybereme jeden náhodný vrchol a sledujme jednu jeho hranu. Otázka pak zní,

<sup>12</sup>Pravděpodobnost  $p_{ij}$  by měla označovat spíše očekávaný počet hran mezi vrcholy  $i$  a  $j$  a ne pravděpodobnost existence hrany, protože jedině pak bude umožněn vznik multihran. Znovu ale platí, že v rozsáhlých grafech se bude počet multihran a smyček blížit k nule.

---

**Algoritmus 4** Generování náhodného grafu s očekávanou stupňovou distribucí.

---

**Require:**  $G, K$ **Ensure:**  $E(G) = \emptyset$  $m \leftarrow (\sum k_i \in K)/2$ **for all**  $v_i, v_j \in V(G); i \leq j$  **do** $x \leftarrow \text{RANDOM}(x \in \mathbb{R}; 0 \leq x < 1)$ **if**  $i \neq j$  **then** $p = \frac{k_i k_j}{2m}$ **else** $p = \frac{k_i^2}{4m}$ **end if****if**  $p > x$  **then** $E(G) \leftarrow E(G) \cup \{(v_i, v_j)\}$ **end if****end for**

---

jaká je pravděpodobnost, že vrchol na konci hrany bude stupně  $k$ ? U konfiguračního modelu byla uvedena podmínka, aby každé možné spojení pahýlů hran bylo stejně pravděpodobné, což je pro tento výpočet podstatné. V grafu je  $2m - 1$  pahýlů pro napojení. Námí vybraný vrchol se může po sledované hraně napojit na  $k$  pahýlů vrcholu stupně  $k$ , přičemž vrcholů stupně  $k$  je celkem  $np_k$ . S možností ignorovat  $-1$  pro velké sítě je pravděpodobnost, že vrchol na konci hrany vedoucí z vybraného vrcholu bude stupně  $k$

$$\frac{k}{2m} np_k = \frac{kp_k}{\langle k \rangle}, \quad (37)$$

kde  $p_k$  je pravděpodobnost, že náhodně vybraný vrchol je stupně  $k$  a  $2m = \langle k \rangle n$ . Excess degree distribution má navíc podmínku, aby v pravděpodobnosti nebyla zahrnuta hrana, kterou byl vrchol dosažen. Tedy pokud se dotazujeme na *excess degree*  $k$ , musí mít soused  $k + 1$  pahýlů včetně toho, kterým jsme se do vrcholu dostali. Toho lze dosáhnout jednoduše záměnou  $k$  za  $k + 1$  ve vztahu 37. Pravděpodobnost  $q_k$ , že náhodný soused vybraného vrcholu bude mít *excess degree*  $k$ , je tak [8, 14]

$$q_k = \frac{(k + 1)p_{k+1}}{\langle k \rangle}. \quad (38)$$

### 3.2.2 Shlukovací koeficient

Shlukovací koeficient konfiguračního modelu lze vypočítat pomocí excess degree distribution z předchozí podkapitoly 3.2.1.

Začneme v libovolném vrcholu  $v$  stupně  $k > 2$  a vyberme náhodně dva jeho sousedy – vrcholy  $i$  a  $j$ . Shlukovací koeficient se ptá, jaká je pravděpodobnost, že  $i$  a  $j$  jsou také spojeny hranou. Pravděpodobnost počtu hran  $k_i$  a  $k_j$  vrcholů  $i$  a  $j$  kromě hrany vedoucí do vrcholu  $v$  lze vypočítat

podle vzorce 38<sup>13</sup>. Vrcholy  $i$  a  $j$  pak mají  $k_i k_j$  možností vzniku hrany a pravděpodobnost vzniku této hrany je podle vzorce 34 rovna  $\frac{k_i k_j}{2m}$ . Sumou přes všechny možné stupně  $k_i$ ,  $k_j$  dostaneme vztah pro výpočet shlukovacího koeficientu [14]:

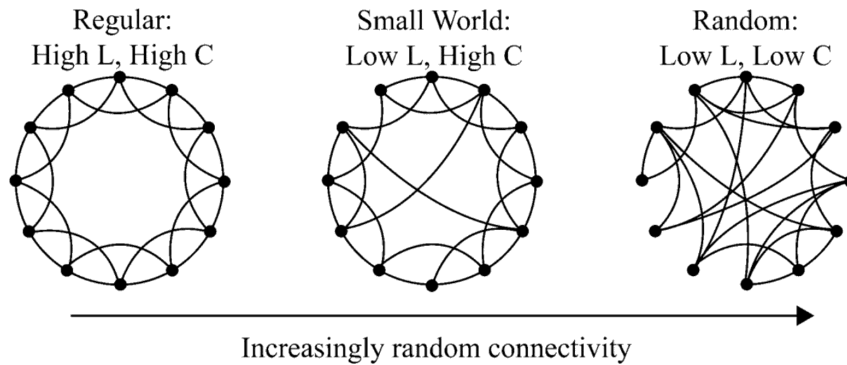
$$C = \sum_{k_i=0} \sum_{k_j=0} q_{k_i} q_{k_j} \frac{k_i k_j}{2m} = \dots = \frac{1}{n} \frac{[\langle k^2 \rangle - \langle k \rangle]^2}{\langle k \rangle^3}, \quad (39)$$

kde  $n$  je počet vrcholů,  $\langle k \rangle$  je průměrný stupeň a  $\langle k^2 \rangle$  je druhý obecný moment vypočtený jako  $\langle k^2 \rangle = \sum_{k=0}^{\infty} k^2 p_k$ .

### 3.3 Modely malého světa

Z předchozího textu vyplývá, že náhodný graf má pro některé reálné sítě nevhodnou distribuci stupňů (pokud nepoužijeme zobecněný náhodný graf) a příliš nízký shlukovací koeficient. Jen nízký průměr náhodného grafu může odpovídat reálným sítím. Tak se dostáváme k modelu, který se snaží vygenerovat graf, jehož shlukovací koeficient by se přiblížil reálným sítím a zachoval by si malý průměr. Tomuto modelu se říká *model malého světa* nebo *Watts-Strogatz model* (dále v textu také WS model) podle jmen jeho tvůrců<sup>14</sup>, kteří jej popsali v roce 1998. Kromě tohoto modelu malého světa existuje mírně odlišný *Watts-Newman model* (dále také WN model). Od těchto modelů se dá očekávat, že v porovnání s náhodným grafem získaným ze stejných parametrů, bude jejich shlukovací koeficient viditelně vyšší.

Základní idea vychází z faktu, že pravidelná (trojúhelníková) mřížka má vysoký shlukovací koeficient, ale také velký průměr, a náhodný graf má naopak malý průměr, ale také nízký shlukovací koeficient. Budeme tak vycházet z mřížky a přepojíme (Watts-Strogatz model) nebo přidáme (Watts-Newman model) pár hran. Těmto přepojeným (přidaným) hranám se říká *zkratky*<sup>15</sup> Tak zůstane zachován vysoký shlukovací koeficient, ale podstatně se sníží průměr grafu [15].



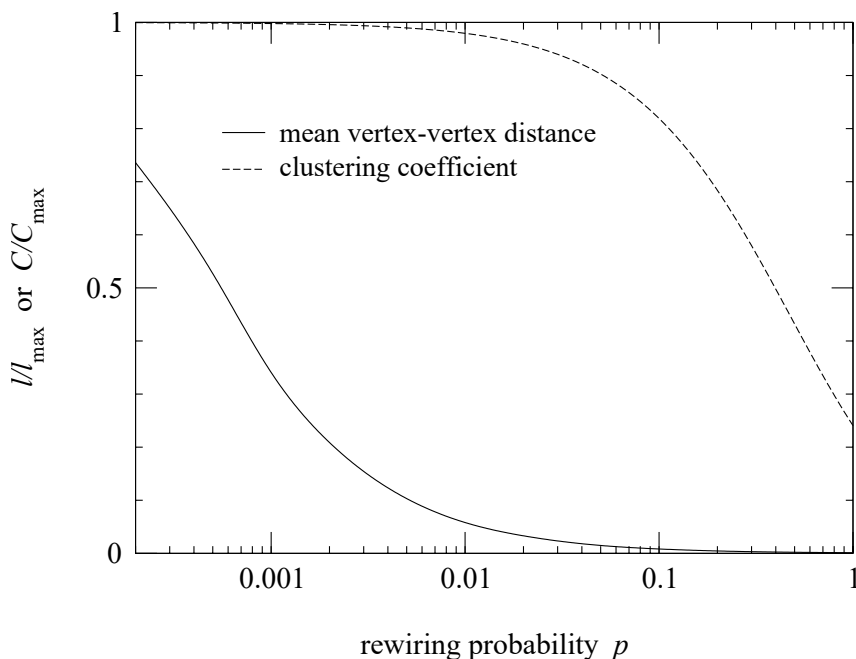
Obrázek 6: Změna vlastností grafu s rostoucím  $p$ , zdroj [6].

<sup>13</sup>Pravděpodobnost počtu hran  $k_i$  a  $k_j$  vrcholů  $i$  a  $j$  kromě hrany vedoucí do vrcholu  $v$  je tedy rozdělena podle „excess degree distribution“.

<sup>14</sup>Duncan Watts a Steve Strogatz.

<sup>15</sup>Přímý překlad anglického „shortcuts“.

Nejčastěji se vychází z jednorozměrné kruhové mřížky s  $n$  vrcholy, jak je vidět na obrázku 6 vlevo. V této mřížce se každý vrchol připojí k nejbližším  $k$  vrcholům a to směrem vpravo i vlevo [15, 20]. U tohoto modelu  $k$  neznačí stupeň vrcholu, který je roven  $2k$ . Proces přepojování (přidávání) hran tvořících zkratky prochází kruhovou mřížku hranu po hraně a s pravděpodobností  $p$  přepojí jeden konec této hrany k náhodně vybranému vrcholu, přičemž nesmí být vytvořeny smyčky nebo multihrany. Pokud je  $p = 0$ , pak zůstává kruhová mřížka s velkým shlukovacím koeficientem (pro  $k \rightarrow \infty$  je  $\langle C \rangle = \frac{3}{4}$ , viz kapitola 3.3.2) a velkým průměrem. Pokud  $p = 1$ , graf se změní v náhodný graf s malým shlukovacím koeficientem a průměrem. Znázornění je na obrázku 6. To, co potřebujeme, se děje mezi těmito mezními hodnotami  $p$ . S rostoucím  $p$  se průměr snižuje podstatně rychleji než shlukovací koeficient a vytváří se vhodný interval hodnot  $p$  jak lze vidět na obrázku 7.



Obrázek 7: Průběh průměru a shlukovacího koeficientu WS modelu v závislosti na  $p$ . Mezi krajními hodnotami je interval s vysokým shlukovacím koeficientem a malým průměrem, zdroj [15].

V případě Watts-Strogatz i Watts-Newman modelu začíná generování vytvořením pravidelné mřížky s  $k$  vazbami k nejbližším vrcholům. Jeden z možných postupů je naznačen algoritmem 5.

Rozdíly mezi WS a WN modelem jsou v následujícím postupu. Watts-Strogatz model přepojuje hrany s pravděpodobností  $p$ . Také neumožňuje vznik multihran a smyček<sup>16</sup>. Cílem Watts-Newman modelu je odstranit některé problémy WS modelu [15, 16]. Hrany jsou s pravděpodobností  $p$  přidávány a je explicitně umožněn vznik multihran a smyček [16]. O přidání hrany je rozhodováno pro každou možnou dvojici vrcholů, ale je procházena původní mřížka a pro každou hranu v mřížce je rozhodováno, zda se propojí dva náhodně vybrané vrcholy. Děje se tak pro zachování kompatibility s WS modelem, aby průměr počtu přidávaných (přepojených)

<sup>16</sup>Z matematických důvodů se model někdy zjednodušuje aby umožňoval vznik smyček a multihran [15].

---

**Algoritmus 5** Generování jednorozměrné kruhové mřížky.

---

**Require:**  $G, k$ **Ensure:**  $0 \leq 2k \leq |V(G)| - 1$ **Ensure:**  $E(G) = \emptyset$  $iterace \leftarrow k$ **while**  $iterace \leq k$  **do****for all**  $v_i \in V(G)$  **do** $E(G) \leftarrow E(G) \cup \left\{ \left\{ v_i, v_{((i+iterace) \bmod |V(G)|)} \right\} \right\}$  $\triangleright$  indexy  $i$  vrcholů začínají od nuly**end for** $iterace \leftarrow iterace + 1$ **end while**

---

hran byl u obou modelů roven  $nkp$  [15, 16]. Možný postup pro generování Watts-Strogatz a Watts-Newman modelu je popsán algoritmy 6 a 7. V některých případech je možné jako základ pro generování modelu zvolit i vícerozměrnou mřížku [16].

---

**Algoritmus 6** Generování grafu podle Watts-Strogatz modelu

---

**Require:**  $G, p$ **Ensure:**  $0 \leq p \leq 1$ **Ensure:**  $G$  je mřížka vygenerovaná algoritmem 5**Ensure:**  $G$  není kompletní graf $H \leftarrow G$ **for all**  $v_i \in V(G)$  **do****for all**  $v_j \in V(G); \{v_i, v_j\} \in E(G)$  **do** $x \leftarrow \text{RANDOM}(x \in \mathbb{R}; 0 \leq x < 1)$ **if**  $p > x$  **then****do** $v_k \leftarrow \text{RANDOM}(v \in V(H))$ **while**  $v_k = v_i$  **or**  $\{v_i, v_k\} \in E(H)$  $E(H) \leftarrow E(H) \setminus \{\{v_i, v_j\}\}$  $E(H) \leftarrow E(H) \cup \{\{v_i, v_k\}\}$ **end if****end for****end for** $G \leftarrow H$ 

---

### 3.3.1 Průměrný stupeň a distribuce stupňů

Pro WS model průměrný stupeň závisí pouze na  $k$  spojeních k nejbližším vrcholům a je roven  $2k$ . U WN modelu se navíc přičítají i stupně z přidanych hran a dostáváme vztah pro průměrný stupeň

$$\langle k \rangle = \begin{cases} 2k & \text{pro Watts-Strogatz model,} \\ \frac{2kn+2knp}{n} = 2k(1+p) & \text{pro Watts-Newman model.} \end{cases} \quad (40)$$



---

**Algoritmus 7** Generování grafu podle Watts-Newman modelu

---

**Require:**  $G, p$ **Ensure:**  $0 \leq p \leq 1$ **Ensure:**  $G$  je mřížka vygenerovaná algoritmem 5**Ensure:**  $G$  není kompletní graf $m \leftarrow E(G)$ **while**  $m > 0$  **do** $x \leftarrow \text{RANDOM}(x \in \mathbb{R}; 0 \leq x < 1)$ **if**  $p > x$  **then** $v_i \leftarrow \text{RANDOM}(v \in V(H))$  $v_j \leftarrow \text{RANDOM}(v \in V(H))$  $E(G) \leftarrow E(G) \cup \{\{v_i, v_j\}\}$ **end if** $m \leftarrow m - 1$ **end while**

---

Distribuce stupňů modelů malého světa neodpovídá většině reálných sítí, což ale ani nebylo záměrem těchto modelů. Pro WN model má každý vrchol stupeň nejméně  $2k$  (z kruhové mřížky) a počet zkratk se řídí binomickým rozdělením. Pak pravděpodobnost, že náhodně vybraný vrchol bude stupně  $j$ , je [15]

$$\Pr(j) = \begin{cases} 0 & \text{pokud } j < 2k, \\ \binom{n}{j-2k} \left(\frac{2kp}{n}\right)^{j-2k} \left(1 - \frac{2kp}{n}\right)^{n-j+2k} & \text{jinak.} \end{cases} \quad (41)$$

Tento výraz je stále poměrně snadno odvoditelný. Naopak odvození výrazu pro WS model je podstatně složitější [5]

$$\Pr(j) = \begin{cases} 0 & \text{pokud } j < k, \\ \sum_{m=0}^{\min(j-k, k)} \binom{k}{m} (1-p)^m p^{k-m} \frac{(kp)^{j-k-m}}{(j-k-m)!} e^{-pk} & \text{jinak.} \end{cases} \quad (42)$$

Distribuce stupňů těchto modelů nepatří mezi známé rozdělení pravděpodobnosti, nicméně její průběh se podobá distribuci stupňů náhodného grafu (tedy Poissonovu rozdělení).

### 3.3.2 Shlukovací koeficient

Výpočet shlukovacího koeficientu začíná opět kruhovou mřížkou. Úvaha vychází z podílu vazeb mezi sousedními vrcholy vrcholu  $i$  a maximálního počtu těchto vazeb, využívá se tedy vztahu 12. Pěkné matematické odvození je uvedeno v [20]. Jiné odvození může být následující. Stupeň každého vrcholu je roven  $2k$ , a proto je maximální počet vazeb mezi sousedy  $i$ -tého vrcholu  $\binom{2k}{2}$ , čímž dostáváme jmenovatel vztahu. Výpočet počtu vazeb mezi sousedními vrcholy vrcholu  $i$  lze rozdělit do tří částí. Pokud jsou oba sousedé vpravo od vrcholu  $i$  je počet vazeb mezi těmito vrcholy  $\binom{k}{2}$ . Totéž platí pokud jsou oba sousední vrcholy vlevo. Ještě zbývá možnost, kdy je jeden

soused vlevo a druhý vpravo. Pokud je sousední vrchol ve vzdálenosti  $l$  od vrcholu  $i$  směrem vlevo, pak má  $k - l$  vazeb na sousedy vrcholu  $i$  směrem vpravo. To vede na sumu

$$\sum_{l=1}^k (k-l) = k^2 - \frac{k(k+1)}{2} = \frac{k(k-1)}{2}. \quad (43)$$

Vztah pro výpočet shlukovacího koeficientu mřížky je pak

$$\langle C \rangle = \frac{\binom{k}{2} + \binom{k}{2} + \frac{k(k-1)}{2}}{\binom{2k}{2}} = \frac{3k(k-1)}{2k(2k-1)} = \frac{3(k-1)}{2(2k-1)}. \quad (44)$$

U Watts-Strogatz modelu dále platí, že pokud při  $p = 0$  byly dva sousední vrcholy vrcholu  $i$  spojeny hranou, pak při  $p > 0$  jsou tyto dva vrcholy stále spojeny s pravděpodobností  $1 - p$  a stále jsou oba vrcholy sousedy vrcholu  $i$  s pravděpodobností  $(1 - p)^2$ . Průměrný shlukovací koeficient WS modelu je [5]

$$\langle C \rangle = \frac{3(k-1)}{2(2k-1)} (1-p)^3. \quad (45)$$

Pro Watts-Newman model je průměrný shlukovací koeficient [15]

$$\langle C \rangle = \frac{3(k-1)}{2(2k-1) + 4kp(p+2)}. \quad (46)$$

### 3.3.3 Průměrná nejkratší cesta

Průměrnou nejkratší cestu modelů malého světa lze poměrně snadno vypočítat pro mezní hodnoty pravděpodobnosti  $p$  přepojení (přidání) hrany jako [5]

$$\langle l \rangle \sim \begin{cases} \frac{n(n+2k-2)}{4k(n-1)} \sim \frac{n}{4k} & \text{pokud } p = 0, \\ \frac{\log(n)}{\log(2k-1)} \simeq \frac{\log n}{\log(k)} & \text{pokud } p = 1. \end{cases} \quad (47)$$

Určení průměrné nejkratší cesty pro libovolné hodnoty parametru  $p$  je obtížné, nicméně její hodnotu lze získat ze vztahu [15]

$$\langle l \rangle = \frac{n}{k} f(nkp), \quad (48)$$

kde funkce  $f(x)$  je přibližně

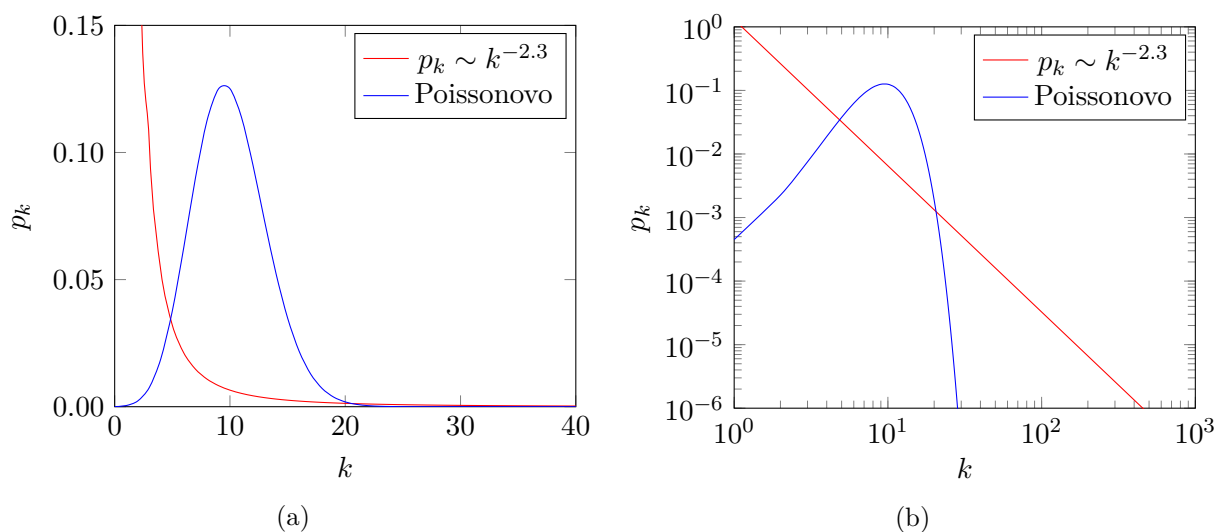
$$f(x) = \frac{1}{2\sqrt{x^2+2x}} \tanh^{-1} \sqrt{\frac{x}{x+2}}. \quad (49)$$

## 3.4 Modely bezškálových sítí – Barabási-Albert Model

Poslední uvedenou kategorií jsou modely generující takzvané „scale-free“ sítě. Konkrétně bude zmíněn jeden, pravděpodobně nejznámější model této kategorie *Barabási-Albert Model*. Všechny předchozí modely, s výjimkou zobecněného náhodného grafu, generovaly sítě mající Poissonovo

rozdělení stupňů vrcholů nebo v případě modelů malého světa distribuci stupňů podobnou Poissonovu rozdělení. Z pozorování reálných sítí ale vyplývá, že často reálné sítě nemají takovou distribuci stupňů. V „lidských sítích“, jako například sociální sítě (sítě přátelství) nebo WWW, se setkáváme s distribucí stupňů podle power-law<sup>17</sup>, jak již bylo uvedeno v kapitole 2.2.4. Platí, že bezškálová síť je síť, jejíž distribuce stupňů se řídí mocninným zákonem (power-law) [4]. Porovnání Poissonova rozdělení a power-law je na obrázku 8. Graf rozdělení pravděpodobnosti podle power-law se často zobrazuje v log-log měřítku, čímž vznikne přímka, jejíž směrnice je rovna exponentu  $-\alpha$  mocninného zákona.

Co to znamená, když je síť bezškálová? V sítích s Poissonovým rozdělením stupňů bude stupeň náhodně vybraného vrcholu v okolí průměrného stupně  $\langle k \rangle$  a tento průměrný stupeň tak slouží jako takzvaná „škála“. Směrodatná odchylka pro Poissonovo rozdělení je dána  $\sigma = \sqrt{\langle k \rangle}$ , a proto bude náhodně vybraný vrchol s velkou pravděpodobností<sup>18</sup> stupně  $\langle k \rangle \pm \sqrt{\langle k \rangle}$  [4]. Lze tak odhadnout, že libovolně vybraný vrchol bude mít stupeň blízký průměrnému stupni  $\langle k \rangle$ . Bezškálové sítě takovou škálu postrádají, pokud je exponent mocninného rozdělení  $\alpha < 3$ , protože pak směrodatná odchylka pro  $n \rightarrow \infty$  diverguje [4]. V praxi to znamená, že u bezškálových sítí není možné odhadnout stupeň náhodně vybraného vrcholu, tedy že stupeň náhodného vrcholu je  $\langle k \rangle \pm \infty$  a v síti existují vrcholy značně odlišných stupňů. To má za následek vznik takzvaných *center*, což jsou vrcholy velkého stupně.



Obrázek 8: Porovnání Poissonova rozdělení a power-law. Pro power-law je mocninný exponent  $\alpha = 2.3$ , a pro Poissonovo rozdělení počet vrcholů  $n = 1000$  a průměrný stupeň  $\langle k \rangle = 10$ , který tvoří „škálu“. Na obrázku (b) porovnání v log-log měřítku, kde lze v případě power-law vidět přímku, jejíž směrnice je rovna  $-\alpha$ .

<sup>17</sup>Nicméně narazit na čistou formu power-law tak, jak je popsána v tomto dokumentu, můžeme v reálném světě jen zřídka [4].

<sup>18</sup>Za předpokladu normálního rozdělení s pravděpodobností přibližně 68%.

*Barabási-Albert Model* model (dále označován také jako BA model) přistupuje k problému generování ve srovnání s předchozími modely odlišně. V reálném světě se často setkáváme s bezškálovými sítěmi a model předpokládá, že takovou síť vytvoří, když se mu podaří napodobit její vznik. Představme si nějakou sociální síť, například spolek prominentních lidí, kde vrcholy jsou lidé a hrany přátelství. Ze začátku je síť malá a tvoří ji jen zakládající členové. Časem se přidá pár dalších lidí a je logické, že se budou chtít přátelit i se dříve vstupujícími členy, kteří se tak začínají stávat ve spolku známými lidmi. Jak vstupují další členové tak vzniká velká šance, že budou znát alespoň některé známé lidi ve spolku, kteří se tak stanou ještě známějšími. Tito nejznámější lidé ve spolku pak v síti tvoří centra. Z postupu vzniku takové sítě se dají vypočítat dva důležité principy a sice *růst*, kdy se přidávají nové vrcholy, a *preferenční připojování*, což znamená, že nové vrcholy mají tendenci se připojovat k vrcholům vyššího stupně.

Barabási-Albert model je založen na tomto růstu a preferenčním připojování. BA model je definován následujícím způsobem. Na začátku máme  $m_0$  vrcholů, mezi nimiž jsou hrany umístěny libovolně s podmínkou, aby každý vrchol měl alespoň jednu hranu. Pak přichází na řadu *růst* a v každém kroku se přidá nový vrchol stupně  $m \leq m_0$  a připojí k  $m$  již existujícím vrcholům v grafu<sup>19</sup>. Podle *preferenčního připojování* je pravděpodobnost  $p_i$ , že se nový vrchol připojí k vrcholu  $i$  dána jako

$$p_i = \frac{k_i}{\sum_j k_j}, \quad (50)$$

kde  $k_i$  je stupeň  $i$ -tého vrcholu [3]. Po  $t$  krocích BA model vygeneruje graf, který má  $m_0 + t$  vrcholů a  $m_0 - 1 + mt$  až  $\binom{m_0}{2} + mt$  hran v závislosti na počátečním grafu.

Ve výsledku tedy BA model generuje bezškálové síť a je založen na dvou principech – *růstu* a *preferenčním připojování*. Distribuce stupňů ve vygenerované síti se pak řídí mocninným zákonem (viz kapitola 2.2.4) s exponentem  $\alpha = 3$  [3]. Příklad možného algoritmu je naznačen pseudokódem 8.

### 3.4.1 Průměrný stupeň a distribuce stupňů

Průměrný stupeň se dá obecně vypočítat jednoduše pomocí vztahu 3 (strana 17) využívajícího princip sudosti jako

$$\langle k \rangle = \frac{2M}{N}, \quad (51)$$

kde  $M$  je počet hran v grafu a  $N$  počet vrcholů.<sup>20</sup> Jak již bylo řečeno, počet vrcholů  $N$  bude po  $t$  krocích  $N = m_0 + t$  a počet hran  $M$  bude v intervalu  $M \in \langle m_0 - 1 + mt, \binom{m_0}{2} + mt \rangle$ , kde  $m_0$  je počet vrcholů počátečního grafu. BA model neklade žádné omezení na podobu tohoto počátečního grafu a obecně se  $m_0$  volí v řádu jednotek a většinou je  $m_0 \ll t$ . Pak můžeme pro

<sup>19</sup>Poslední částí souvětí je zabráněno vzniku smyček a multihran.

<sup>20</sup>Doposud byl v práci značen počet hran  $m$  a počet vrcholů  $n$ . U BA modelu ale  $m$  značí počet hran, kterými bude nový vrchol připojen k síti, a proto byly malé písmena ve vzorci nahrazeny velkými.

---

**Algoritmus 8** Generování grafu podle Barabási-Albert modelu

---

**Require:**  $G, n, m, m_0$

**Ensure:**  $G$  je obyčejný graf na  $m_0$  vrcholech, kde každý vrchol  $v$  je stupně  $k_v \geq 1$ .

**Ensure:**  $m \leq m_0$

$D[]$  = pole, kde každý vrchol  $v \in V(G)$  je v poli právě  $k_v$  krát, kde  $k_v$  je stupeň vrcholu  $v$ .

**for**  $i \leftarrow m_0$  **to**  $n - 1$  **do**

$degree \leftarrow 0$

$degreeSum \leftarrow SIZE(D[])$

    ▷ Velikost pole odpovídá součtu stupňů vrcholů.

**while**  $degree < m$  **do**

$x \leftarrow RANDOM(x \in \mathbb{N}; 0 \leq x < degreeSum)$

**if**  $E(G) \cap \{\{v_i, D[x]\}\} = \emptyset$  **then**

        ▷ Nebyl znovu vybrán stejný vrchol.

$E(G) \leftarrow E(G) \cup \{\{v_i, D[x]\}\}$

        ▷  $D[x]$  provede výběr vrcholu podle vzorce 50.

$D[].APPEND(v_i)$

$D[].APPEND(D[x])$

$degree \leftarrow degree + 1$

**end if**

**end while**

**end for**

---

rozsáhlé sítě vliv  $m_0$  ignorovat. Průměrný stupeň  $\langle k \rangle$  BA modelu je pak [3]

$$\langle k \rangle = \frac{2M}{N} = \frac{2mt}{t} = 2m. \quad (52)$$

Odvodit přesný vztah pro distribuci stupňů BA modelu je složitější, nicméně podle [3] je dán jako

$$\Pr(k) = \frac{2m(m+1)}{k(k+1)(k+2)}. \quad (53)$$

Z tohoto vztahu lze vypočítat hned několik skutečností. Zaprvé se pro velké  $k$  vztah „zredukuje“ a dostáváme  $p_k \sim k^{-3}$ , což odpovídá zmiňovanému mocninnému zákonu  $p_k \sim k^{-\alpha}$  s exponentem  $\alpha = 3$ . Dále je tento exponent nezávislý na parametru  $m$ . Je tedy jedno, jaké  $m$  zvolíme. Zajímavé také je, že distribuce stupňů BA modelu, na rozdíl od náhodných grafů, nezávisí na velikosti sítě (počtu vrcholů). Z toho plyne, že pravděpodobnost výběru vrcholu stupně například 50 je stejná pro síť se 100 vrcholy i pro síť s  $10^{1000}$  vrcholy.

### 3.4.2 Shlukovací koeficient

Odvodit analytický výpočet shlukovacího koeficientu BA modelu je složitější než v případě předchozích modelů. Samotné odvození vede na vztah [3]

$$C = \frac{m(\ln N)^2}{4N}. \quad (54)$$

Z něj vyplývá úměrnost  $C \sim \frac{(\ln N)^2}{N}$ , což vede na vyšší shlukovací koeficient než byl v případě náhodného grafu ( $C \sim \frac{1}{N}$ ). Nicméně z pozorování reálných sítí vyplývá, že jejich průměrný

shlukovací koeficient je nezávislý na počtu vrcholů a lokální shlukovací koeficient je závislý na stupni vrcholu [2]. BA model tak nemusí být schopný zachytit shlukovací koeficient reálných sítí.

### 3.4.3 Průměrná nejkratší cesta

Průměrnou nejkratší cestu Barabási-Albert modelu, kde stupeň každého nově přidaného vrcholu  $m \geq 2$ , lze vypočítat vztahem [3]

$$\langle l \rangle \sim \frac{\log n}{\log \log n}. \quad (55)$$

Z toho plyne, že pro  $n \rightarrow \infty$  bude průměrná nejkratší cesta  $\langle l \rangle$  BA modelu menší než v případě náhodného grafu (za předpokladu konstantního průměrného stupně  $\langle k \rangle$ ).

## 4 Implementace aplikace pro generování sítí

Součástí práce je aplikace umožňující generovat grafy (sítě) mající požadované vlastnosti. Prvním důležitým krokem při tvorbě je dobře si rozmyslet požadavky na aplikaci.

Aplikace bude konzolová a umožní podle parametrů zadaných pomocí přepínačů generovat grafy s různými vlastnostmi vycházejícími z modelů zmíněných v první části práce, konkrétně jsou to tyto modely:

**Náhodný graf** z kapitoly 3.1 a to podle  $G(n, p)$  i  $G(n, m)$  definice. Vstupem tak bude počet vrcholů  $n$  a pravděpodobnost  $p$  spojení dvou vrcholů nebo počet vrcholů a počet hran  $m$ . Vznik smyček ani multihran podle algoritmů 1 a 2 nebude umožněn. U tohoto modelu jako jediného bude umožněno generovat orientovaný graf a ohodnocený graf s náhodně generovanými váhami hran ze zadaného intervalu.

**Zobecněný náhodný graf** z kapitoly 3.2 a to jak *náhodný graf s očekávanou stupňovou posloupností* neboli *konfigurační model*, tak i *náhodný graf s očekávanou stupňovou distribucí*. Vstupem bude posloupnost stupňů vrcholů nebo počet vrcholů  $n$  a mocninný exponent  $\alpha$ , z čehož bude vygenerována posloupnost stupňů vrcholů podle mocninného zákona. Vznik smyček a multihran bude pro konfigurační model umožněn. Pro náhodný graf s očekávanou stupňovou distribucí bude umožněn pouze vznik smyček.

**Modely malého světa** z kapitoly 3.3, konkrétně *Watts-Strogatz* a *Watts-Newman* model. Vstupem bude počet vrcholů  $n$ , počet sousedů každého vrcholu  $k$  a pravděpodobnost přepojení/přidání hrany  $p$ . Watts-Strogatz model nebude umožňovat vznik smyček a multihran, Watts-Newman ano.

**Barabasi-Albert model** z kapitoly 3.4. Vstupem bude počet vrcholů  $n$ , počet vrcholů počátečního grafu  $m_0$  a počet hran nově přidaného vrcholu  $m$ . Vznik smyček a multihran nebude umožněn.

Jako programovací jazyk byl zvolen jazyk C# založený na platformě .NET Framework. Díky tomu je možné využít kvalitní knihovnu tříd (Class library).

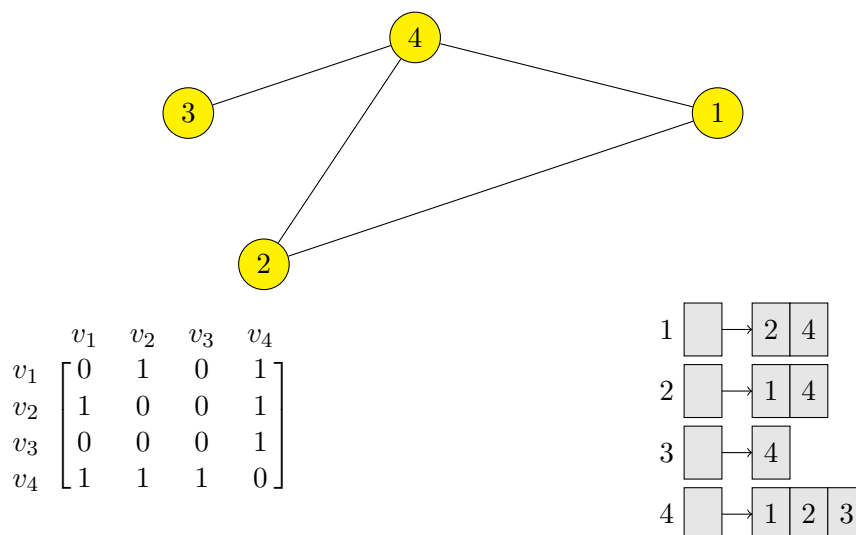
Každý vygenerovaný graf půjde vyexportovat do souboru *.csv* tak, aby na každém řádku byla dvojice vrcholů spojených hranou oddělená středníkem, tedy ve formátu „ $v_i; v_j$ “. V případě ohodnocených grafů ve formátu „ $v_i; v_j; w_{ij}$ “, kde  $w_{ij}$  je váha hrany. Seznam ID všech vrcholů bude v dalším *.csv* souboru, každé ID bude na jednom řádku. Při exportu bude možné přepínačem ignorovat smyčky a multihrany, smyčky tak budou přeskočeny a multihrany nahrazeny jednoduchou hranou. Dále bude aplikace umožňovat pro vygenerované grafy změřit některé jejich základní vlastnosti: *průměr*, *průměrnou nejkratší cestu*, *lokální shlukovací koeficient*, *průměrný shlukovací koeficient*, *průměrný vstupní/výstupní stupeň*, *hustotu*, dále *počet komponent*<sup>21</sup> a pro orientovaný graf *počet silně/slabě souvislých komponent*, a v neposlední řadě

<sup>21</sup>Včetně získání kolekce obsahující jednotlivé komponenty.

také *distribuci stupňů* a pro orientovaný graf *distribuci vstupních/výstupních stupňů*. Některé z těchto vypočítaných vlastností budou následně uloženy do souboru *.txt*. Do něj bude také kromě změřeného uložena „očekávaný“ průměrný stupeň a shlukovací koeficient, oba vypočítané ze vstupních parametrů modelu podle analytických vzorců zmíněných v první části práce. Distribuce stupňů vrcholů vygenerovaného grafu bude také uložena do samostatného souboru *.csv* ve formátu „stupeň;počet“, kde *počet* je počet vrcholů daného *stupně*.

#### 4.1 Použité struktury

Jak víme, síť lze reprezentovat pomocí grafů a prvním důležitým rozhodnutím před implementací aplikace je zvolit správný způsob reprezentace grafů v paměti počítače. Můžeme se setkat se dvěma nejběžnějšími způsoby, a sice s reprezentací pomocí *matice sousednosti* nebo *seznamem vrcholů a jejich sousedů*. Matice sousednosti si v poli o rozměrech  $n \times n$ , kde  $n$  je počet vrcholů, ukládá na pozici  $ij$  informaci o tom, zda jsou vrcholy  $v_i$  a  $v_j$  spojeny hranou. Seznam sousedů si pro každý vrchol  $i$  ukládá seznam vrcholů, do kterých vede hrana z vrcholu  $i$ . Na obrázku 9 je příklad grafu, pro který je graficky znázorněna reprezentace pomocí matice sousednosti a seznamu sousedů.



Obrázek 9: Grafické znázornění matice sousednosti a seznamu sousedů pro jednoduchý neorientovaný graf.

Ačkoliv mohou být některé operace s maticí sousednosti jednodušší, její největší nevýhoda je vyšší paměťová náročnost  $O(n^2)$ . Oproti tomu paměťová náročnost seznamu sousedů je  $O(n+m)$ . Tento rozdíl není z obrázku 9 příliš patrný, nicméně pro neorientovaný jednoduchý graf, který má počet vrcholů  $n = 1000$  a průměrný stupeň  $\langle k \rangle = 4$ , je rozdíl  $10^6$  prvků v matici oproti 5000 položkám v seznamu vrcholů a jejich sousedů. Jak bylo zmíněno v kapitole 2.2.4, hustota reálných sítí je malá, tedy  $m \ll n$ . Proto bude reprezentace seznamem sousedů lepší volbou.



Protože aplikace má umožňovat práci s orientovanými/neorientovanými ohodnocenými/neohodnocenými multigrafy, je dobré jednoduchou představu seznamu sousedů pro tyto typy grafů zpřesnit. Neorientovaný neohodnocený multigraf může být reprezentován přesně tak, jak je znázorněno na obrázku 9. Ohodnocené multigrafy mají navíc váhu hrany a ta může být uložena v seznamu sousedů společně s ID sousedního vrcholu. U orientovaných multigrafů můžeme rozlišovat *následovníky* vrcholu  $v_i$  (tedy vrcholy, do kterých vede hrana z vrcholu  $v_i$ ), *předchůdce* vrcholu  $v_i$  (tedy vrcholy, ze kterých vede hrana do vrcholu  $v_i$ ) a *sousedy* vrcholu  $v_i$ , kteří vzniknou sjednocením předchůdců a následovníků. U orientovaných multigrafů tak může být vhodné kromě seznamu následovníků ukládat i seznam předchůdců.

## 4.2 Návrh aplikace

V této kapitole budou společně s diagramy a ukázkami kódu velmi stručně popsány základní části aplikace. Základ celé aplikace tvoří třída *Graph*, viz obrázek 10, která zprostředkovává základní operace nad grafem a udržuje graf v paměti pomocí seznamu sousedů.

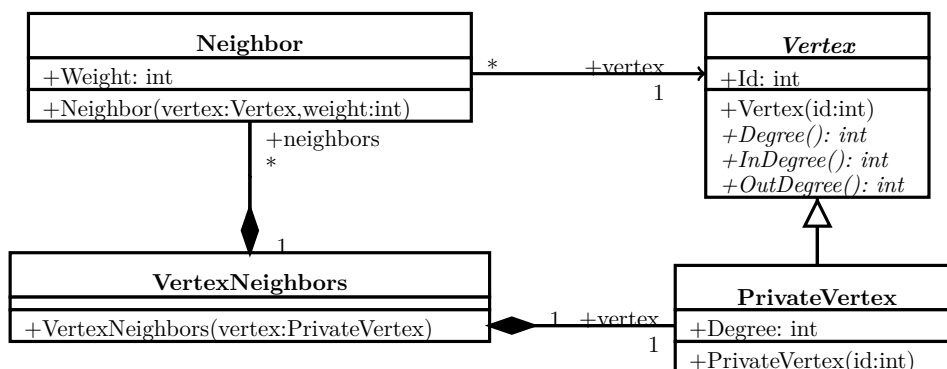
Graph
+Vertices: int +Edges: int +IsDirected: bool +IsWeighted: bool #succesors: VertexNeighbors[*]
+Graph(vertices:int,weighted:bool) #InitVertices(numOfVertices:int) +AddEdge(v_i:int,v_j:int): void +AddEdge(v_i:int,v_j:int,weight:int): void +RemoveEdge(v_i:int,v_j:int): void +IsEdge(v_i:int,v_j:int): bool +GetVertex(id:int): Vertex +GetEdge(v_i:int,v_j:int): IEnumerable<Edge> +GetVertices(): IEnumerable<Vertex> +GetEdges(ignoreMultiedges:bool,ignoreSelfLoops:bool): IEnumerable<Edge> +GetNeighbors(v_i:int): IEnumerable<Neighbor> +GetSuccesors(v_i:int): IEnumerable<Neighbor> +GetPredecessors(v_i:int): IEnumerable<Neighbor> +GetSubGraph(vertices:IEnumerable<int>): Graph

Obrázek 10: Třída Graph.

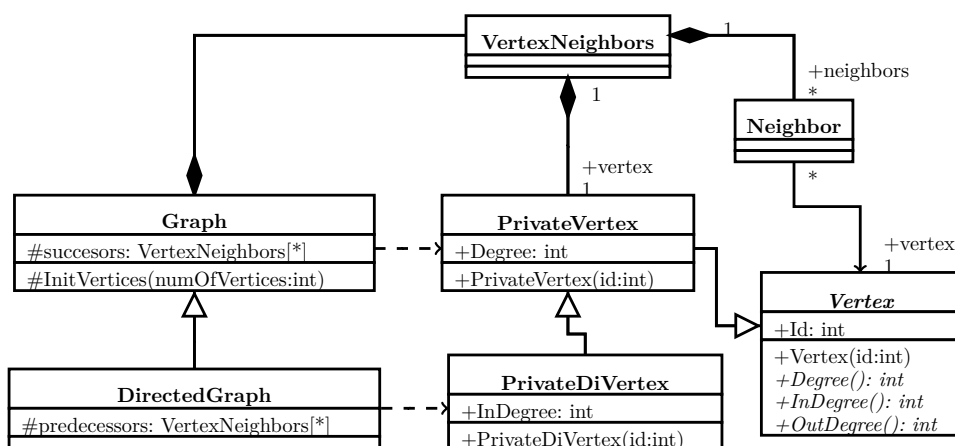
Seznamy sousedů obsahuje pole *succesors* typu *VertexNeighbors*, jež se inicializuje v metodě *InitVertices* na potřebný počet vrcholů. Struktura seznamu sousedů je naznačena na obrázku 11.

Třída *VertexNeighbors* má seznam typu *Neighbor* reprezentující samotný seznam sousedů. Třída *Neighbor*, představující jednoho souseda, má váhu hrany vedoucí k tomuto sousedovi a vrchol souseda. Každý vrchol má *Id* shodující se s indexem v poli *succesors* třídy *Graph* a umí poskytnou svůj stupeň v závislosti na tom, zda je součástí orientovaného nebo neorientovaného grafu. Pro neorientovaný graf je vrchol představován třídou *PrivateVertex*, která vhodně implementuje abstraktní metody vracející stupeň vrcholu<sup>22</sup>. Pro orientovaný graf je vrchol představován třídou *PrivateDiVertex*, jak je ukázáno na obrázku 12.

<sup>22</sup>Někdo by mohl namítnout, že udržovat si u vrcholu jeho stupeň je zbytečné a lze jej snadno zjistit z počtu



Obrázek 11: Třídní diagram s třídami pro reprezentaci seznamu sousedů.



Obrázek 12: Třídní diagram zachycující vztah orientovaného a neorientovaného grafu.

Třída *DirectedGraph* vhodným způsobem překrývá potřebné metody včetně *InitVertices*, kde inicializuje seznamy předchůdců a následovníků, jak je ukázáno ve výpisu 1.

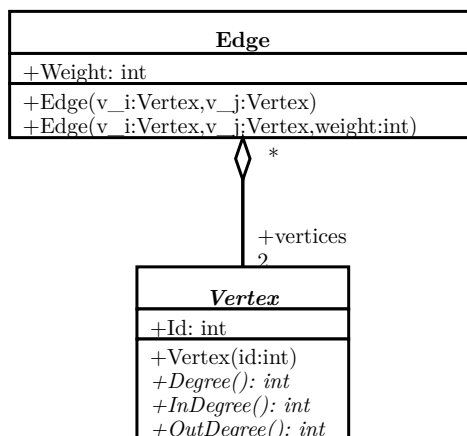
```
protected override void InitVertices(int numOfVertices)
{
    succesors = new VertexNeighbors[numOfVertices];
    predecessors = new VertexNeighbors[numOfVertices];

    for (int i = 0; i < numOfVertices; i++)
    {
        PrivateDiVertex v = new PrivateDiVertex(i);
        succesors[i] = new VertexNeighbors(v);
        predecessors[i] = new VertexNeighbors(v);
    }
}
```

Výpis 1: Inicializace seznamů předchůdců a následovníků.

Pro zjednodušenou představu o základu aplikace sloužící k reprezentaci grafu zbývá ještě doplnit obrázek 13 s třídou *Edge* představující hranu mezi vrcholy.

položek v seznamu sousedů. To je pravda, problém ale nastává u smyček, kdy se pro neorientovaný graf počet položek v seznamu zvýší o 1, ale stupeň vrcholu o 2. Při dotazu na stupeň vrcholu by pak bylo nutné celý seznam sousedů projít a spočítat smyčky, což by zvýšilo časovou složitost dotazu.



Obrázek 13: Reprezentace hrany grafu v aplikaci.

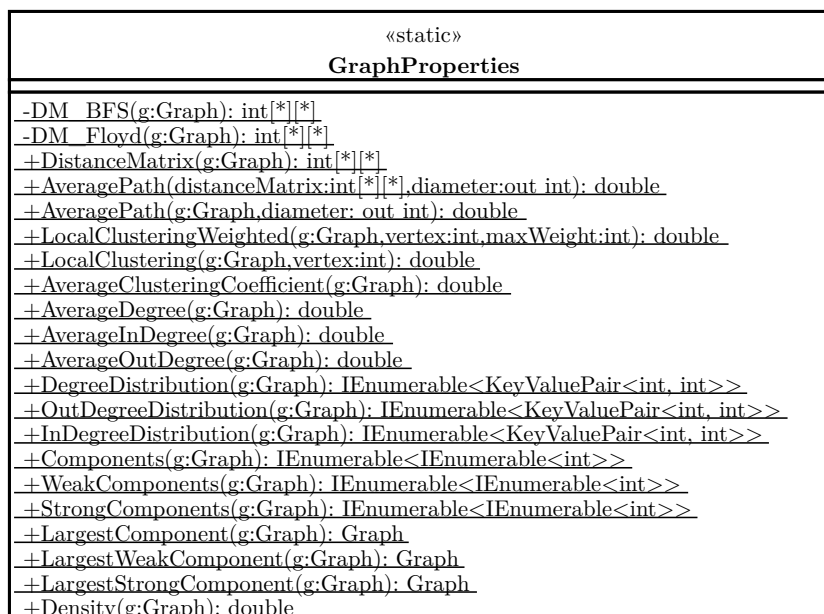
Tabulka 1: Asymptotické časové složitosti metod třídy *Graph* v nejhorším případě.

Metoda	Asymptotická složitost
AddEdge	$O(1)$
RemoveEdge	$O(n)^*$
IsEdge	$O(n)^*$
GetVertex	$O(1)$
GetEdge	$O(n)^*$
GetVertices	$O(n)^*$
GetEdges	$O(n^2)^*$
GetNeighbors	$O(1)$
GetSuccessors	$O(1)$
GetPredecessors	$O(1)$
GetSubgraph	$O(n^2)$

\* Pouze v případě kompletního grafu.

Představit si konkrétní implementaci ostatních metod třídy *Graph* by nemělo být příliš složité, a proto nemá smysl v textu uvádět další ukázky kódu přímo související s touto třídou. Namísto ukázek kódu bude vhodnější uvést tabulku 1 shrnující asymptotické složitosti jednotlivých metod třídy *Graph*.

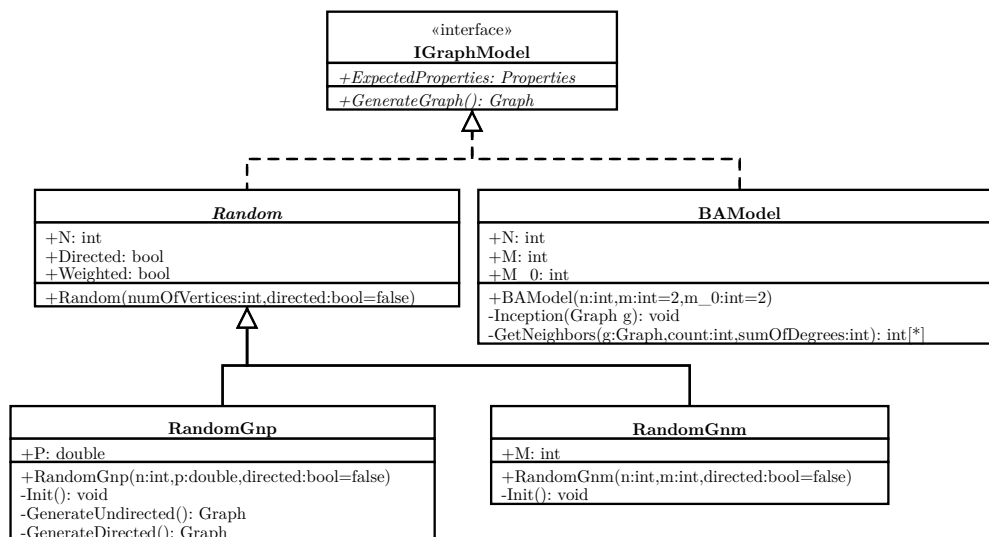
Pro výpočty některých vlastností grafů slouží statická třída *GraphProperties* znázorněna na obrázku 14, kterou můžeme považovat za „knihovnu“ funkcí. Metoda *DistanceMatrix* pouze volá jednu z privátních metod *DM\_BFS* nebo *DM\_Floyd* v závislosti na tom, zda je graf ohodnocený nebo neohodnocený. Metody *(In/Out)DegreeDistribution* vrací seřazenou kolekci dvojic klíčů a hodnot, kde klíčem je stupeň vrcholu a hodnotou počet vrcholů daného stupně. Metody *(Weak/Strong)Components* vrací kolekci obsahující seznamy ID vrcholů patřících do jedné komponenty. Největší komponenta tak bude reprezentována nejdelším seznamem v kolekci. Účel ostatních metod třídy *GraphProperties* by měl být patrný z jejich názvu.



Obrázek 14: Třída *GraphProperties* pro výpočty vlastností grafu.

Implementační náročnost jednotlivých metod se liší, nicméně většina těch složitějších je založena na známých algoritmech, jako jsou prohledávání do šířky nebo prohledávání do hloubky. Použité algoritmy a asymptotická složitost metod třídy *GraphProperties* jsou shrnuty v tabulce 2.

Následuje popis části programu sloužící k samotnému generování sítě podle modelů uvedených v práci. Všechny modely jsou reprezentovány vlastní třídou implementující rozhraní *IGraphModel*, jak je pro některé naznačeno na obrázku 15.



Obrázek 15: Rozhraní modelu.

Tabulka 2: Asymptotické časové složitosti metod třídy *GraphProperties* pro výpočty nad grafem a algoritmy, na kterých jsou tyto metody založeny. Asymptotické časové složitosti byly určeny z implementovaných algoritmů.

Metoda	Algoritmus	Asymptotická složitost
DM_BFS	BFS	$O(n \times (n + m))$
DM_Floyd	Floyd	$O(n^3)$
AveragePath	-	$O(n^2)$
LocalClusteringWeighted	-	$O(n^2)$
LocalClustering	-	$O(n^2)$
AverageClusteringCoefficient	-	$O(n^3)$
Average(In/Out)Degree	-	$O(n)$
(In/Out)InDegreeDistribution	-	$O(n)$
(Weak)Components	BFS	$O(n + m)$
StrongComponents	Tarjan(DFS) [18]	$O(n + m)$
Largest(Strong/Weak)Component	-	$O(n^2)$
Density	-	$O(n)$

Každý model umí také spočítat některé svoje očekávané vlastnosti, konkrétně průměrný stupeň a shlukovací koeficient, které jsou analyticky vypočítány ze vzorců uvedených v první části textu (kapitola 3). Tyto vlastnosti jsou uloženy v *ExpectedProperties*. Protože se hlavní část práce zabývá modely sítí, je jejich implementace včetně ukázek kódů detailněji rozepsána v následující kapitole.

K exportu grafu slouží statická třída *GraphExporter*, která exportuje graf do souboru *.csv*. Příklad postupu vytvoření a uložení grafu podle  $G(n, m)$  modelu je zachycen sekvenčním diagramem na obrázku 25 v přílohách.

### 4.3 Implementace algoritmů modelů

V této kapitole bude podrobněji popsána konkrétní implementace algoritmů pro generování modelů komplexních sítí. Celkem je implementováno 7 modelů.

Prvními jsou dva generující náhodný graf z kapitoly 3.1. Jejich implementace podle algoritmů 1 a 2 je poměrně přímočará a jednoduše modifikovatelná pro orientované grafy. Pro zájemce je jejich implementace uvedena v přílohách ve výpisech 7 a 8. Následují dva modely generující zobecněný náhodný graf a sice *konfigurační model* a *náhodný graf s očekávanou stupňovou distribucí*. *Náhodný graf s očekávanou stupňovou distribucí* lze znovu velmi jednoduše implementovat modifikací algoritmu 4, konkrétní implementace je v přílohách ve výpisu 9. Při implementaci algoritmu pro *konfigurační model* podle pseudokódu 3 lze využít vlastnosti návrhu aplikace umožňující dotazovat se na stupeň libovolného vrcholu grafu v konstantním čase. Tak odpadá potřeba vytvořit si pole o velikosti součtu stupňů vrcholů, kde každý vrchol v tomto poli je právě tolikrát, jaký je jeho stupeň, tedy tak, jak je uvedeno ve zmíněném pseudokódu. Generování konfiguračního modelu tak může probíhat například podle kódu ve výpisu 2.

---

```

public override Graph GenerateGraph()
{
    System.Random rnd = new System.Random();
    Graph g = new Graph(N);

    List<Vertex> verticesList = g.GetVertices().ToList();
    while (verticesList.Count > 0)
    {
        Vertex V1 = verticesList[rnd.Next(verticesList.Count)];
        Vertex V2;
        do
        {
            V2 = verticesList[rnd.Next(verticesList.Count)];
        } while ((V1.Id == V2.Id) && (V1.Degree == DegreeSequence[V1.Id] - 1)); //Pokud
            pracuju s 1 vrcholem, pak je nutne mit v "rezerve" 2 stupne

        g.AddEdge(V1, V2);

        if (V1.Degree == DegreeSequence[V1.Id])
            verticesList.Remove(V1);
        if ((V1.Id != V2.Id) && (V2.Degree == DegreeSequence[V2.Id]))
            verticesList.Remove(V2);
    }

    return g;
}

```

---

Výpis 2: Generování konfiguračního modelu.

Dalšími modely jsou *Watts-Strogatz* a *Watts-Newman* model spadající do skupiny modelů malého světa. Základem pro oba je vygenerování kruhové mřížky podle algoritmu 5. Poté je u *Watts-Strogatz* modelu pro každou hranu v původní mřížce rozhodováno, zda bude přepojena. Pro získání dvojic ID vrcholů tvořících hranu v této mřížce je implementován iterátor, jak je ukázáno ve výpisu 3. Samotný kód pro vygenerování mřížky, *Watts-Strogatz* modelu a *Watts-Newman* modelu je v přílohách ve výpisech 10, 11 a 12.

---

```

protected int GetVertex(int idVertex)
{
    return (idVertex >= 0) ? idVertex % N : N - (idVertex % N);
}

protected IEnumerable<ProtectedEdge> VirtualGrid()
{
    int j = 1;
    while (j <= K)
    {
        for (int i = 0; i < N; i++)
            yield return new ProtectedEdge(GetVertex(i), GetVertex(i + j));
        j++;
    }
}

```

---

Výpis 3: Iterátor kruhové mřížky.

Poslední skupinu tvoří modely bezškálových sítí, do které spadá Barabási-Albert Model. Může se zdát, že nevýhodou algoritmu 8 zmíněném v kapitole 3.4 je udržování nějaké struktury (například pole nebo seznamu) o velikosti součtu stupňů všech vrcholů v grafu, což zvyšuje paměťovou složitost algoritmu. K problému generování s dodržением *preferenčního připojování* lze potom přistoupit i jiným způsobem s konstantní paměťovou složitostí (za cenu zvýšení časové složitosti). Stačí si pamatovat pouze součet stupňů všech vrcholů  $degSum(G)$  a generovat náhodné reálné číslo  $x$  v intervalu  $(0, degSum(G))$ . Poté postupně sčítat stupně vrcholů, dokud

číslo  $x$  nebude spadat do intervalu součtu před přičtením stupně posledního zpracovávaného vrcholu a po přičtení stupně posledního zpracovávaného vrcholu. Experimentálně byly tyto dva možné algoritmy otestovány a bylo zjištěno, že na běžném počítači není v tomto případě vyšší paměťová složitost takovým problémem jako vyšší časová složitost. Z tohoto důvodu bylo rozhodnuto zůstat u původní myšlenky zachycené algoritmem 8. Konkrétní implementace algoritmu pro vygenerování BA modelu je rozdělena do tří částí: vytvoření počátečního grafu na  $m_0$  vrcholech (je zvolen kompletní graf, výpis 4), získání  $m$  vrcholů pro připojení nového vrcholu podle preferenčního připojování (výpis 5) a samotné připojení nového vrcholu ke grafu (výpis 6).

---

```
private void Inception(Graph g, IList<int> vertexArray)
{
    for (int i = 0; i < M_0; i++)
    {
        for (int j = i + 1; j < M_0; j++)
        {
            g.AddEdge(i, j);
            vertexArray.Add(i);
            vertexArray.Add(j);
        }
    }
}
```

---

Výpis 4: Vytvoření počátečního grafu BA modelu.

---

```
private int[] GetNeighbors(int count, System.Random rnd, IReadOnlyList<int> vertexArray)
{
    HashSet<int> neId = new HashSet<int>();
    while (neId.Count < count)
    {
        int dediredIndex = rnd.Next(vertexArray.Count);
        int vertex = vertexArray[dediredIndex];
        neId.Add(vertex);
    }
    return neId.ToArray();
}
```

---

Výpis 5: Získání sousedů pro nový vrchol BA modelu podle preferenčního připojování.

---

```
public Graph GenerateGraph()
{
    System.Random rnd = new System.Random();
    Graph g = new Graph(N);

    List<int> vertexArray = new List<int>(); // seznam, kde kazdy vrchol je tolikrat,
    // jaky je jeho stupen
    Inception(g, vertexArray); // pocatek

    for (int i = M_0; i < N; i++) // rust, pridam nový vrchol, který se pripoji k m <=
    // m_0 vrcholům v grafu
    {
        int[] nei = GetNeighbors(M, rnd, vertexArray); // ziska vrcholy pro pripojeni
        // podle preferencniho pripojovani
        for (int j = 0; j < M; j++)
        {
            g.AddEdge(i, nei[j]);
            vertexArray.Add(i);
            vertexArray.Add(nei[j]);
        }
    }
    return g;
}
```

---

Výpis 6: Generování Barabási-Albert modelu.

## 5 Experimenty nad vygenerovanými daty

V této kapitole budou provedeny experimenty nad vygenerovanými daty. Jako první se nabízí porovnat vygenerované grafy a jejich změřené vlastnosti s vlastnostmi očekávanými, tedy analyticky vypočtenými a uvedenými v kapitole 3. Může se stát, že ne vždy budou hodnoty odpovídat. V tom případě by se mělo dát zdůvodnit, proč se tak nestalo.

Začněme u náhodného grafu, pro který byly vygenerovány čtyři grafy s různým počtem vrcholů  $n$  a pravděpodobností spojení hran  $p$ . Distribuce stupňů těchto grafů je na obrázku 17 srovnána s odpovídajícím Poissonovým rozdělením. Na obrázku 17d je průměrný stupeň vrcholu  $\langle k \rangle \approx 1$ , a proto zde již není vidět pro Poissonovo rozdělení typická „zvonovitá křivka“. Další vlastnosti vygenerovaných grafů jsou v tabulce 3.

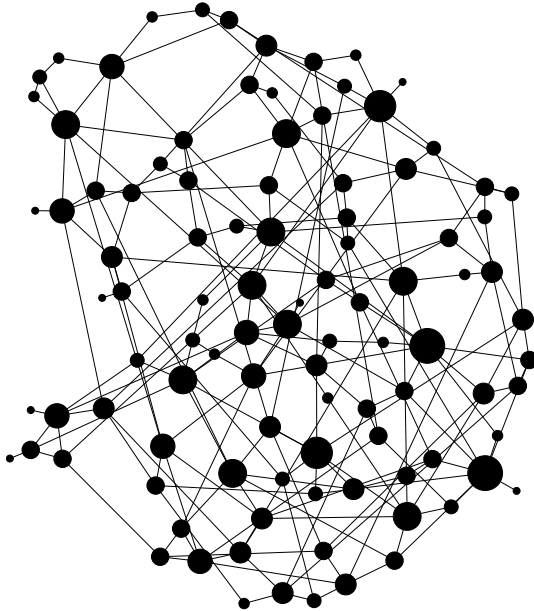
Dalšími testovanými jsou modely zobecněného náhodného grafu z kapitoly 3.2. Znovu jsou vygenerovány čtyři grafy, dva pro *konfigurační model* a dva pro *náhodný graf s očekávanou stupňovou distribucí*, dále označován jako *NG\_DegDist*. Vstupní posloupnost stupňů vrcholů by měla odpovídat power-law a je vygenerována podle vzorce zmíněném v [9] (Apendix D). Distribuce stupňů těchto grafů je na obrázku 18 a ostatní vlastnosti v tabulce 4. V této tabulce by pozornosti neměl ujít extrémně vysoký očekávaný (vypočtený podle vzorce 39) shlukovací koeficient, který je ve dvou případech  $\gg 1$ . Přitom podle definice shlukovací koeficient náleží do intervalu  $(0, 1)$ . Tato „anomálie“ je s největší pravděpodobností způsobena zmíněným generátorem posloupnosti hodnot podle power-law, který neklade omezení na maximální hodnotu v posloupnosti. To jde dobře vidět na obrázku 18a, kde je počet vrcholů  $n = 1000$ , ale vyskytuje se zde jeden vrchol stupně  $k \approx 3000$ . Tak logicky musí v grafu vzniknout značný počet smyček a multihran, jejichž vznik sice konfigurační model umožňuje, ale očekává se, že se jejich počet bude se zvyšujícím se počtem vrcholů blížit nule. Kromě obrázku je tento problém vidět i v tabulce 4, kde je poměrně velký rozdíl mezi očekávaným a skutečným průměrným stupněm u *NG\_DegDist* s  $n = 1000$  a  $\alpha = 2,0$ . Protože *NG\_DegDist* model na rozdíl od konfiguračního modelu neumožňuje vznik multihran, představují tento rozdíl u průměrného stupně právě neexistující multihrany. Pro zvyšující se mocninný exponent  $\alpha$  se pravděpodobnost vzniku „extrémního“ stupně v posloupnosti snižuje a jak je vidět z tabulky, tak pro  $\alpha = 3,0$  již očekávané a skutečné hodnoty odpovídají.

Následuje ověření modelů malého světa. Byly vygenerovány dva grafy podle Watts-Newman modelu a dva grafy podle Watts-Strogatz modelu. Pro porovnání odpovídají vstupní parametry prvním dvěma vygenerovaným náhodným grafům ze začátku kapitoly. Distribuce stupňů těchto grafů je na obrázku 19, ovšem není pro nás až tak zajímavá jako naměřené vlastnosti v tabulce 5. Z tabulky je zřejmé, že pro  $n = 1000$  a  $p = 0,01$  mají oba modely, ve srovnání s náhodným grafem vygenerovaným se stejnou hodnotou parametrů  $n$  a  $p$ , výrazně vyšší shlukovací koeficient. Jejich průměrná cesta je ovšem také mnohem vyšší. Pro hodnoty  $n = 1000$  a  $p = 0,1$  je shlukovací koeficient stále vysoký, ale průměrná cesta se podstatně snížila. Přesně takové chování je od tohoto modelu očekávané a bylo zachyceno na obrázku 7 v kapitole 3.3.

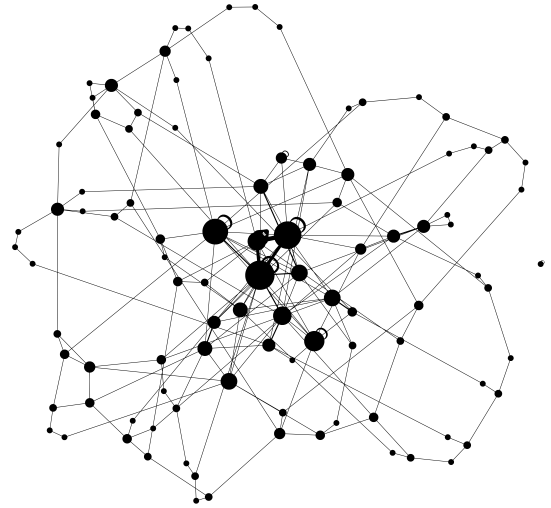


Ověřit ještě zbývá Barabási-Albert model. Znovu byly pro ukázkou vygenerovány čtyři grafy. Distribuce stupňů těchto grafů je na obrázku 20 a některé vlastnosti v tabulce 6.

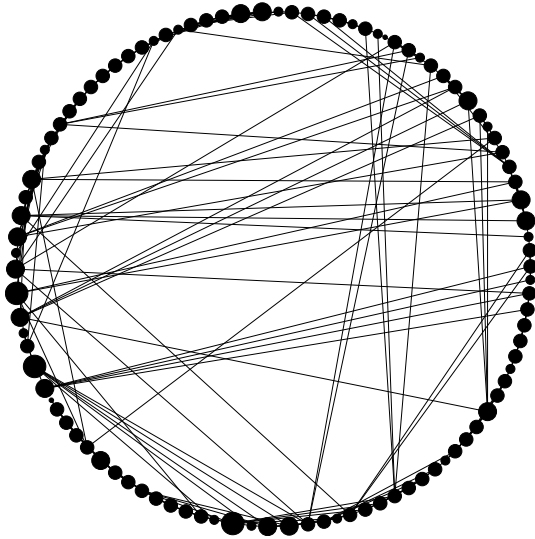
Pro doplnění představy o vygenerovaných grafech je jejich vizualizace programem *Gephi* [21] na obrázku 16.



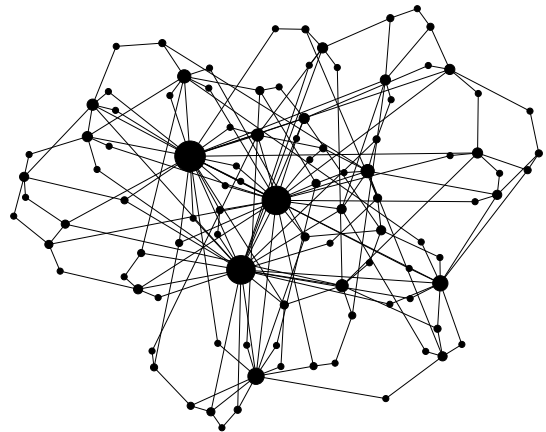
(a) Náhodný graf.



(b) Konfigurační model s posloupností stupňů podle power-law.

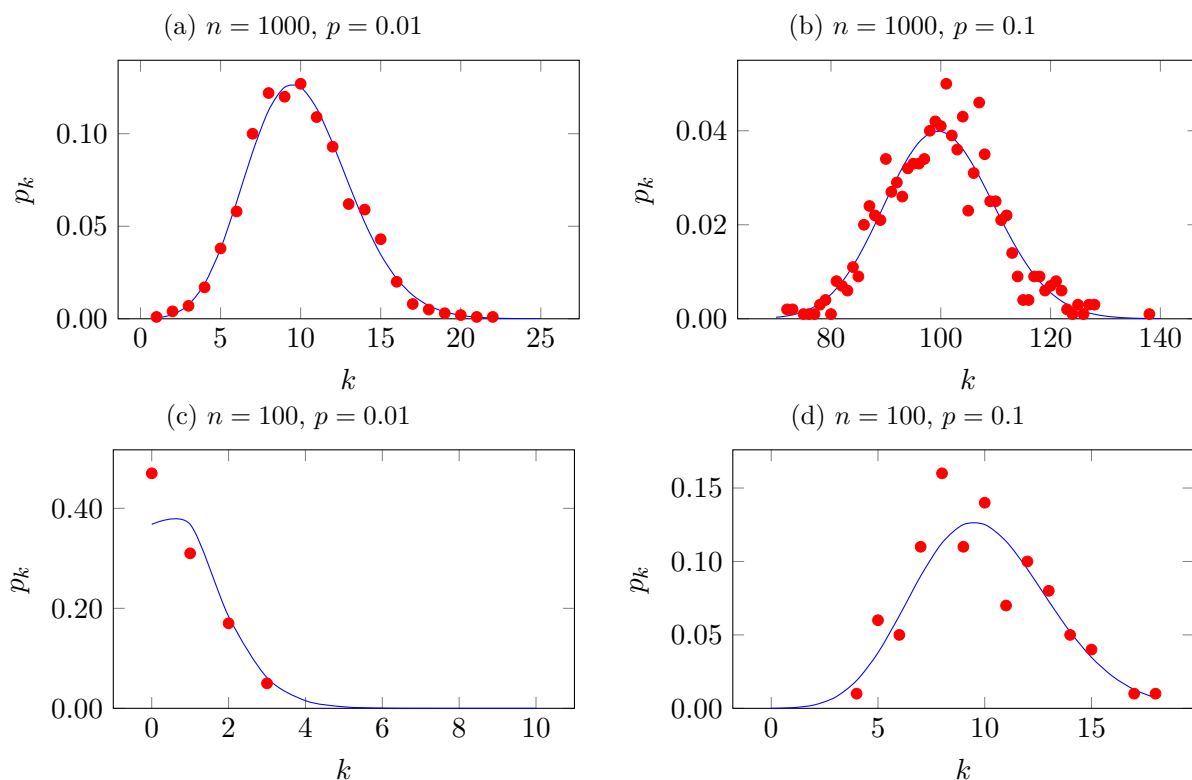


(c) Watts-Strogatz model.



(d) Barabási-Albert model.

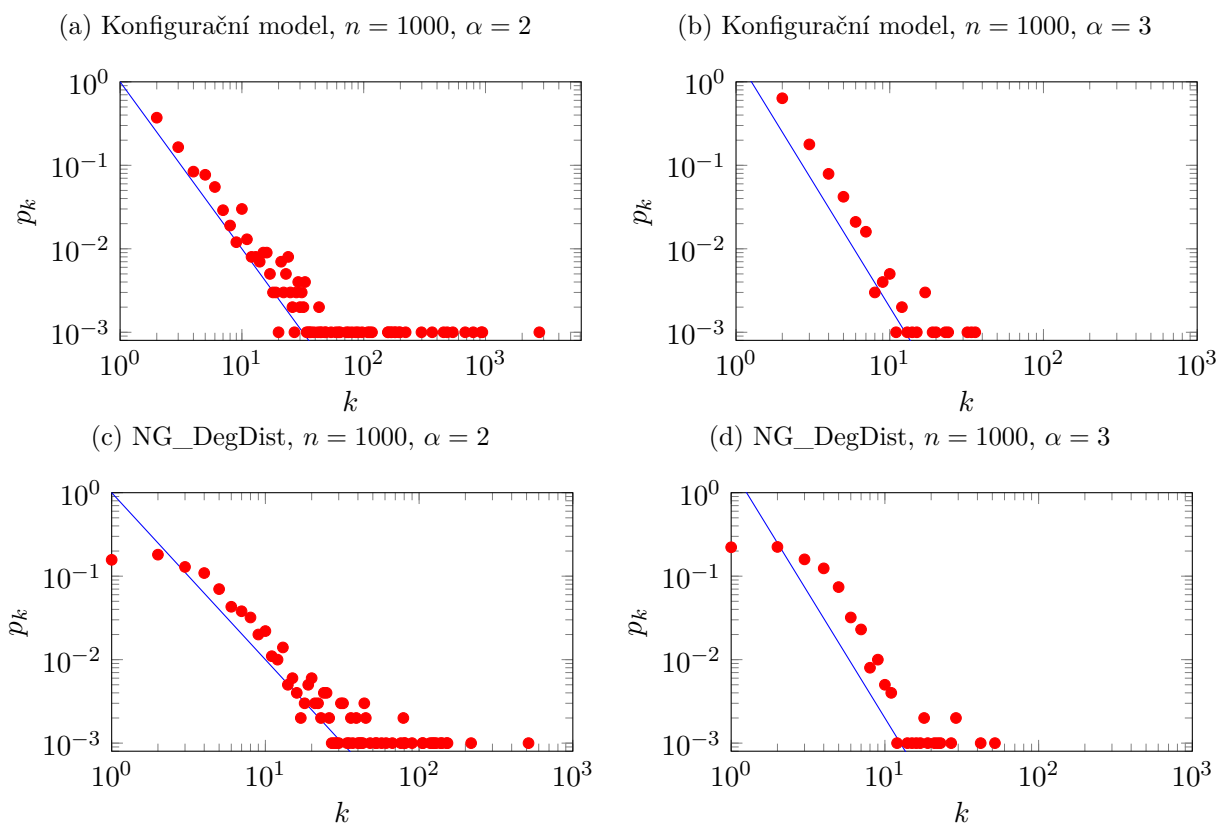
Obrázek 16: Vizualizace vygenerovaných grafů. Všechny grafy mají počet vrcholů  $n = 100$  a průměrný stupeň  $\langle k \rangle \approx 4$ . Velikost vrcholu je úměrná jeho stupni.



Obrázek 17: Distribuce stupňů vygenerovaných náhodných grafů s různými parametry. Modrou křivkou zobrazeno Poissonovo rozdělení.

Tabulka 3: Některé vlastnosti vygenerovaných náhodných grafů s různými parametry.

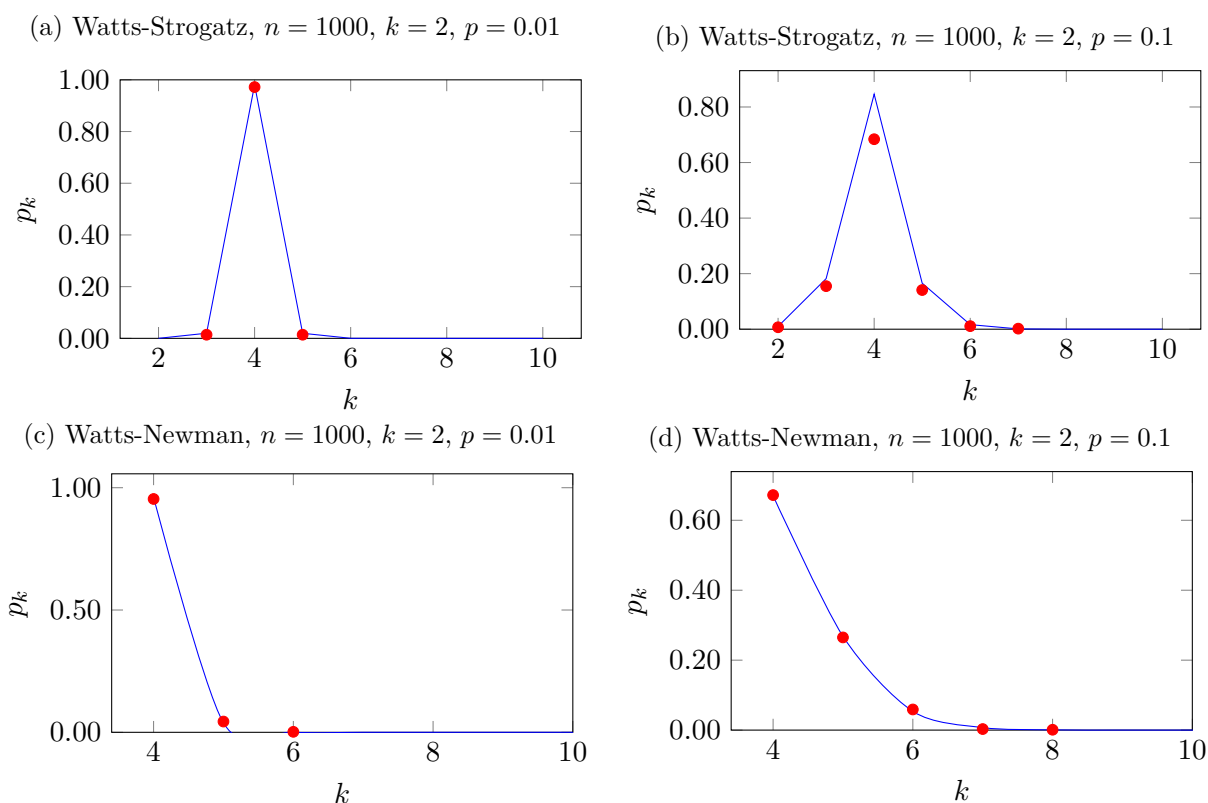
		$n = 1000$ $p = 0,01$	$n = 1000$ $p = 0,1$	$n = 100$ $p = 0,01$	$n = 100$ $p = 0,1$
Průměrný stupeň	Očekávaný	9,99	99,9	0,99	9,9
	Skutečný	9,94	100,12	0,8	9,74
Shlukovací koeficient	Očekávaný	0,01	0,1	0,01	0,1
	Skutečný	0,01	0,1	0	0,1
Hustota		0,01	0,1	0,01	0,1
Komponent		1	1	60	1
Průměr		6	3	7	4
Průměrná cesta		3,26	1,9	2,89	2,25



Obrázek 18: Distribuce stupňů vygenerovaných zobecněných náhodných grafů s různými parametry v log-log měřítku. Modrou křivkou zobrazeno odpovídající mocninové rozdělení.

Tabulka 4: Některé vlastnosti vygenerovaných zobecněných náhodných grafů s různými parametry.

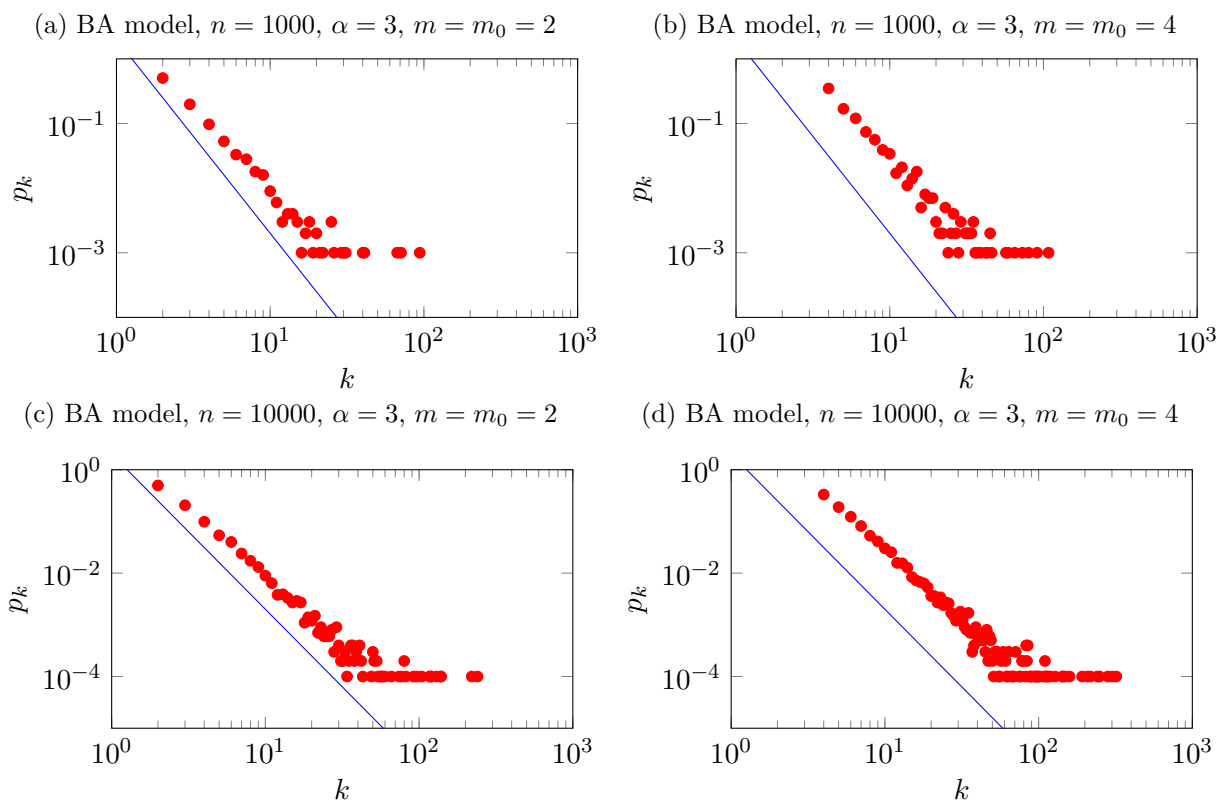
		Konfigurační model		NG_DegDist	
		$n = 1000$ $\alpha = 2,0$	$n = 1000$ $\alpha = 3,0$	$n = 1000$ $\alpha = 2,0$	$n = 1000$ $\alpha = 3,0$
Průměrný stupeň	Očekávaný	16,658	2,996	10,616	3,032
	Skutečný	16,658	2,996	8,036	2,988
Shlukovací koeficient	Očekávaný	30,622	0,007	6,723	0,009
	Skutečný	0,033	0,008	0,333	0,009
Hustota		0,017	0,003	0,008	0,003
Komponent		2	3	66	111
Průměr		9	19	6	13
Průměrná cesta		4,076	7,284	2,673	4,902



Obrázek 19: Distribuce stupňů grafů vygenerovaných podle modelů malého světa. Modrou křivkou zobrazena odpovídající distribuce podle vzorců 41 a 42.

Tabulka 5: Některé vlastnosti vygenerovaných grafů podle modelů malého světa.

		Watts-Strogatz model		Watts-Newman model	
		$n = 1000$ $k = 2$ $p = 0,01$	$n = 1000$ $k = 2$ $p = 0,1$	$n = 1000$ $k = 2$ $p = 0,01$	$n = 1000$ $k = 2$ $p = 0,1$
Průměrný stupeň	Očekávaný	4	4	4,04	4,4
	Skutečný	4	4	4,048	4,396
Shlukovací koeficient	Očekávaný	0,485	0,365	0,487	0,391
	Skutečný	0,488	0,365	0,491	0,429
Hustota		0,004	0,004	0,004	0,004
Komponent		1	1	1	1
Průměr		84	16	66	17
Průměrná cesta		38,201	8,773	25,067	8,203

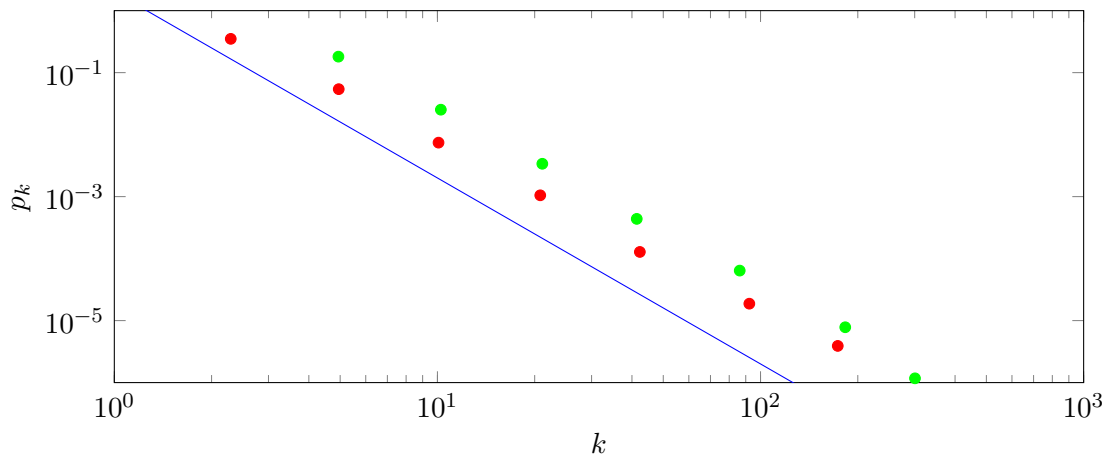


Obrázek 20: Distribuce stupňů grafů vygenerovaných podle Barabási-Albert modelu. Modrou křivkou zobrazeno odpovídající mocninné rozdělení.

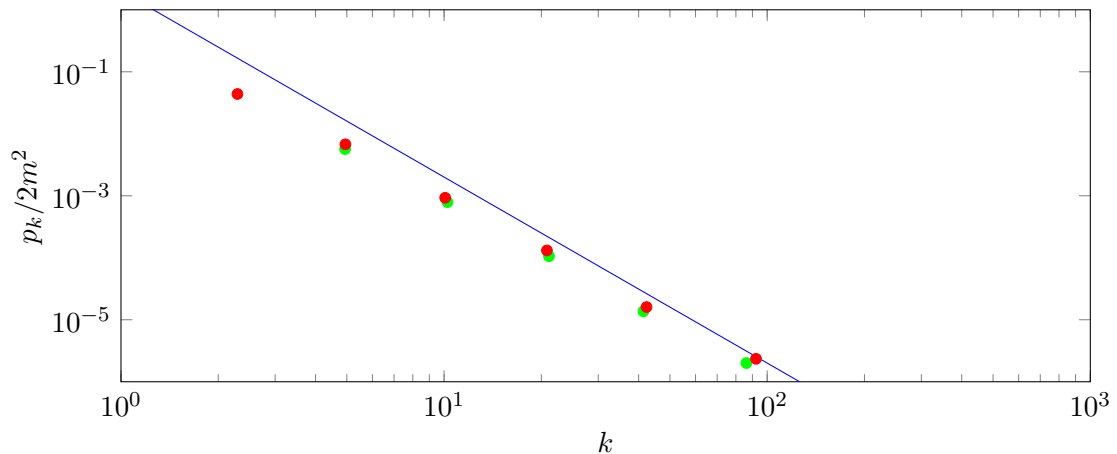
Tabulka 6: Některé vlastnosti vygenerovaných grafů podle Barabási-Albert modelu.

		$n = 1000$	$n = 1000$	$n = 10000$	$n = 10000$
		$\alpha = 3$	$\alpha = 3$	$\alpha = 3$	$\alpha = 3$
		$m = 2$	$m = 4$	$m = 2$	$m = 4$
		$m_0 = 2$	$m_0 = 4$	$m_0 = 2$	$m_0 = 4$
Průměrný stupeň	Očekávaný	4	8	4	8
	Skutečný	3,994	7,98	3,999	7,998
Shlukovací koeficient	Očekávaný	0,024	0,048	0,004	0,008
	Skutečný	0,026	0,031	0,006	0,006
Hustota		0,004	0,008	0,0004	0,0008
Komponent		1	1	1	1
Průměr		7	5	9	6
Průměrná cesta		3,996	3,199	4,998	3,897

Vraťme se ještě ke grafům vygenerovaným podle Barabási-Albert modelu (a také k zobecněnému náhodnému grafu s distribucí stupňů podle power-law). V grafech distribuce stupňů vrcholů může být nepříjemný velký šum na konci (u vrcholů malého stupně). Z takového grafu je obtížné zpětně určit mocninný exponent  $\alpha$ . Jedním z řešení je použít logaritmický „binning“ namísto lineárního [4]. Díky tomu budou „rovnoměrnější“ rozestupy stupňů v grafu s logaritmickým měřítkem. Princip je vysvětlen v [4]. Distribuce stupňů Barabási-Albert modelu z obrázku 20c a 20d s použitím logaritmického „binningu“ je v grafu na obrázku 21. Z tohoto grafu už se dá lépe určit, zda mocninný exponent, tedy sklon přímky, odpovídá očekávání. Z grafu je patrný také určitý „odskok“ distribuce od modré přímky. Ten je způsoben konstantou  $2m(m+1)$  ve vzorci 53 pro distribuci stupňů v BA modelu. Po vydělení pravděpodobnosti  $p_k$  touto konstantou by se již body v grafu měly přiblížit k přímce, jak je na obrázku 22.

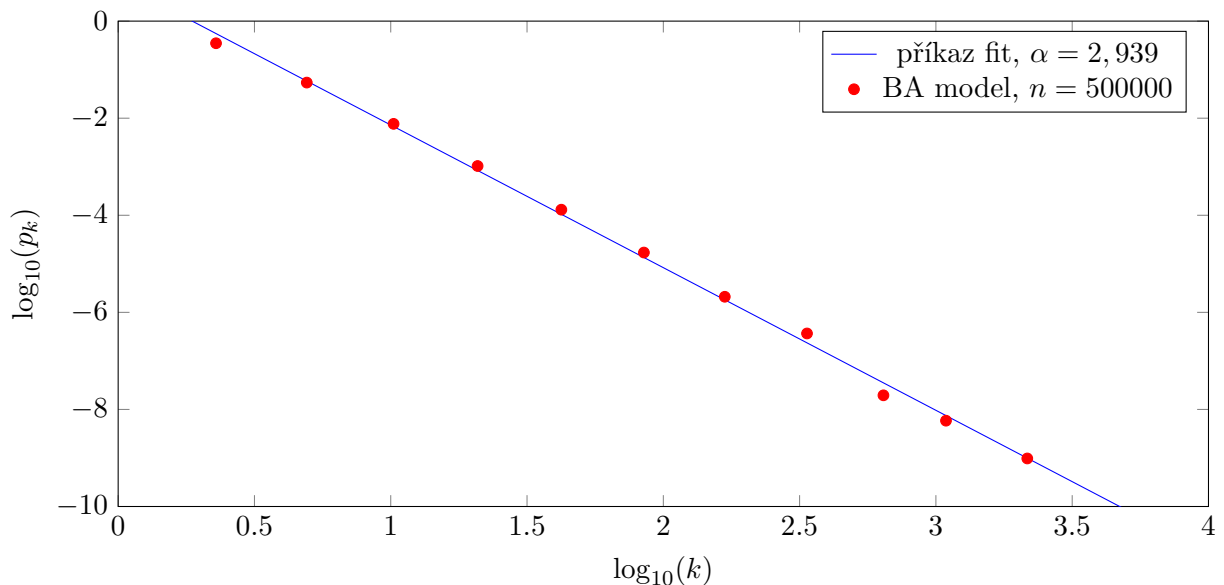


Obrázek 21: Distribuce stupňů grafů BA modelu s použitím logaritmického „binningu“. Počet vrcholů  $n = 10000$ , červeně  $m = m_0 = 2$  a zeleně  $m = m_0 = 4$ .



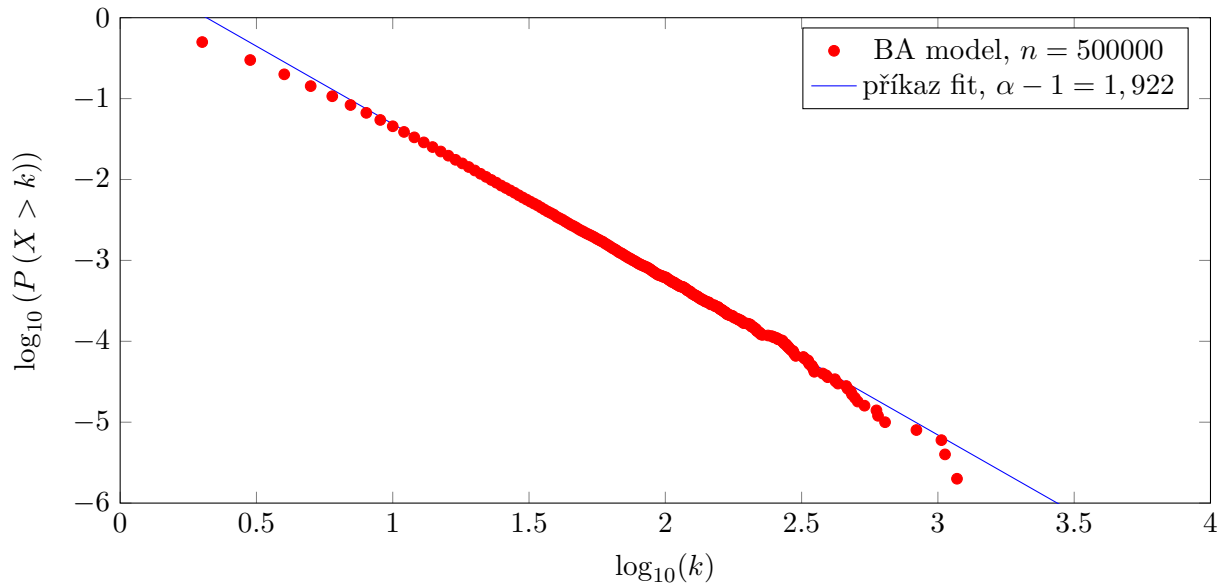
Obrázek 22: Distribuce stupňů grafů BA modelu s použitím logaritmického „binningu“, vydělená  $2m^2$ . Počet vrcholů  $n = 10000$ , červeně  $m = m_0 = 2$  a zeleně  $m = m_0 = 4$ .

Závěrem si můžeme ukázat jeden ze způsobů, jak z existující distribuce stupňů vrcholů bezškálové sítě určit exponent mocninného zákona  $\alpha$ . Toho lze dosáhnout zmíněným logaritmickým „binningem“. Jako modelový příklad poslouží distribuce stupňů grafu s počtem vrcholů  $n = 500000$  vygenerovaného podle BA modelu. U tohoto modelu se očekává exponent mocninného zákona  $\alpha \approx 3$ . Pro samotné určení je použit příkaz *fit* programu *gnuplot* [19]. *Gnuplot* je program pro vykreslování grafů a umí zpracovat jak uživatelsky definované funkce tak i datové soubory. Příkaz *fit* umožňuje data v souboru proložit libovolnou funkcí, jejíž parametry určí metodou nejmenších čtverců. Nejlepších výsledků bylo dosaženo v lineárním měřítku zlogaritmováním hodnot distribuce a funkce mocninného rozdělení. Tímto způsobem byl exponent  $\alpha$  odhadnut na hodnotu  $\alpha = 2,939$ . Výsledek je zobrazen v grafu na obrázku 23.



Obrázek 23: Určení exponentu  $\alpha$  mocninného zákona z distribuce stupňů BA modelu použitím logaritmického „binningu“.

Jiná možnost, jak z existující distribuce stupňů vrcholů bezškálové sítě určit exponent mocninného zákona  $\alpha$ , je použít doplňkovou kumulativní distribuční funkci (CCDF, complementary cumulative distribution function). Pokud vytvoříme doplňkovou kumulativní distribuční funkci pro hustotu pravděpodobnosti spojité náhodné veličiny řídicí se mocninným zákonem (power-law), dostaneme funkci  $\bar{F}(x)$ , která se také řídí mocninným zákonem. Pokud exponent mocninného zákona této funkce  $\bar{F}(x)$  označíme  $\beta$ , pak vztah exponentu  $\beta$  k hledanému exponentu  $\alpha$  bude  $\beta = \alpha - 1$  [9]. Odvození vztahu je uvedeno v příloze F. Pro samotné určení exponentu je znovu použit příkaz *fit* programu *gnuplot*. Pomocí této metody byl exponent  $\alpha$  odhadnut na hodnotu  $\alpha = 2,922$ . Výsledek je zobrazen v grafu na obrázku 24.



Obrázek 24: Určení exponentu  $\alpha$  mocninného zákona z distribuce stupňů BA modelu použitím CCDF.

Provedené experimenty nad vygenerovanými grafy ukázaly, že změřené vlastnosti grafů se prakticky neliší od očekávaných analyticky vypočtených hodnot. Výjimkou byly zobecněné náhodné grafy, kde byl rozdíl způsoben generátorem posloupnosti stupňů vrcholů. Hodnoty u ostatních modelů už odpovídaly očekávání. Na závěr byly otestovány dva způsoby určení exponentu  $\alpha$  mocninného zákona z distribuce stupňů, pomocí kterých byl odhadnut exponent  $\alpha$  mocninného zákona z distribuce stupňů grafu vygenerovaného podle BA modelu.



## 6 Závěr

Hlavním cílem práce bylo prozkoumat problematiku generování sítí s požadovanými vlastnostmi. Tato oblast je mnohem rozsáhlejší, než by se mohlo na první pohled zdát.

Na úvod byly vybrány a popsány některé vlastnosti reálných sítí, které by se daly považovat za nejčastěji zkoumané a nějak související s generováním sítí. V žádném případě je nelze považovat za všechny. Dále bylo analyzováno několik modelů sítí, které bývají v literatuře zmiňovány nejčastěji. Zde vzniká prostor k případnému teoretickému rozšíření práce ve smyslu doplnění zkoumaných vlastností, přidání dalších modelů nebo třeba jen rozšíření BA modelu o orientovanou verzi.

Převážnou část práce věnující se experimentům nad vygenerovanými daty tvoří praktické ověření předpokládaného chování modelů z teoretické části práce, které se dá považovat za úspěšné.

Součástí práce je také implementace aplikace pro vygenerování zmíněných modelů. Ta je konzolová a byla implementovaná v jazyce C#. Kromě generování modelů také umožňuje vypočítat všechny v práci uvedené a zkoumané vlastnosti sítí. Použité konzolové rozhraní aplikace se dá považovat do jisté míry za omezené, protože požadavky na generování sítí mohou být velmi různorodé a je poměrně obtížné všechny v uživatelském rozhraní zachytit, ať je to rozhraní konzolové nebo grafické. Takové uživatelské rozhraní do určité míry omezuje potenciál „jádra“ aplikace. Dalším krokem by mohla být úprava aplikace a celkové rozšíření do podoby jednoduchého balíku (knihovny) pro vytváření, manipulaci a měření vlastností komplexních sítí, který by mohl být použit jako základ pro různé projekty. Samotná aplikace i zdrojové kódy jsou k dispozici na přiloženém DVD.

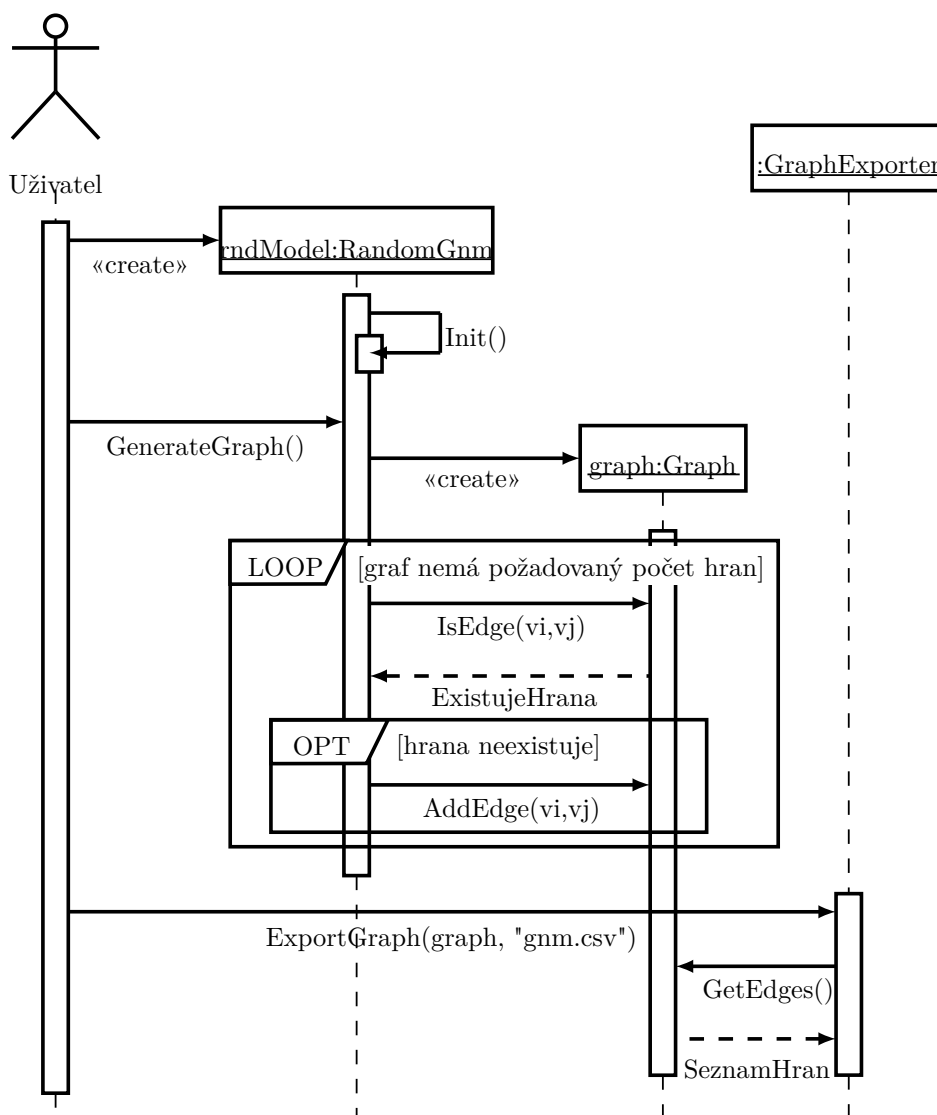
## Literatura

- [1] Barabási, A. *Network Science*. Graph Theory. [online]. [cit 2015-10-10]. Dostupné z: [http://barabasi.com/networksciencebook/content/Chapter2\\_final.pdf](http://barabasi.com/networksciencebook/content/Chapter2_final.pdf)
- [2] Barabási, A. *Network Science*. Random Networks. [online]. [cit 2016-01-15]. Dostupné z: [http://barabasi.com/networksciencebook/content/Chapter3\\_final.pdf](http://barabasi.com/networksciencebook/content/Chapter3_final.pdf)
- [3] Barabási, A. *Network Science*. The Barabási-Albert Model. [online]. [cit 2016-11-13]. Dostupné z: [http://barabasi.com/networksciencebook/content/Chapter5\\_final.pdf](http://barabasi.com/networksciencebook/content/Chapter5_final.pdf)
- [4] Barabási, A. *Network Science*. The Scale-Free Property. [online]. [cit 2016-01-22]. Dostupné z: [http://barabasi.com/networksciencebook/content/Chapter4\\_final.pdf](http://barabasi.com/networksciencebook/content/Chapter4_final.pdf)
- [5] Barrat, A. a M. Weight. On the properties of small-world network models. *The European Physical Journal B* [online]. 2000, **13**(3), 547–560 [cit. 2016-05-07]. DOI: 10.1007/s100510050067. ISSN 1434-6036. Dostupné z: <https://doi.org/10.1007/2Fs100510050067>
- [6] Clauset, Aaron. *Network Analysis and Modeling* [online]. Santa Fe Institute, 2014 [cit. 2016-01-21]. Lecture 1: Introduction and overview. Dostupné z: [http://tuvalu.santafe.edu/~aaronc/courses/5352/csci5352\\_2014\\_L1.pdf](http://tuvalu.santafe.edu/~aaronc/courses/5352/csci5352_2014_L1.pdf)
- [7] Clauset, Aaron. *Network Analysis and Modeling* [online]. Santa Fe Institute, 2014 [cit. 2016-01-25]. Lecture 2: Random graphs I: the Erdos-Renyi model. Dostupné z: [http://tuvalu.santafe.edu/~aaronc/courses/5352/csci5352\\_2014\\_L2.pdf](http://tuvalu.santafe.edu/~aaronc/courses/5352/csci5352_2014_L2.pdf)
- [8] Clauset, Aaron. *Network Analysis and Modeling* [online]. Santa Fe Institute, 2014 [cit. 2016-05-14]. Lecture 4: Random graphs II: the configuration model. Dostupné z: [http://tuvalu.santafe.edu/~aaronc/courses/5352/csci5352\\_2014\\_L4.pdf](http://tuvalu.santafe.edu/~aaronc/courses/5352/csci5352_2014_L4.pdf)
- [9] Clauset, Aaron, Cosma Rohilla SHALIZI a M. E. J. NEWMAN. Power-Law Distributions in Empirical Data. *SIAM Review* [online]. 2009, **51**(4), 661–703 [cit. 2017-02-14]. DOI: 10.1137/070710111. ISSN 0036-1445. Dostupné z: <http://epubs.siam.org/doi/10.1137/070710111>
- [10] Demlová, Marie. *Diskrétní matematika a logika pro KM* [online]. FEL ČVUT, 2010 [cit. 2017-01-13]. Dostupné z: <http://math.feld.cvut.cz/demlova/teaching/dmc/dohromady10.pdf>
- [11] Fagiolo, Giorgio. Clustering in complex directed networks. *Physical Review E* [online]. 2007, **76**(2), [cit. 2017-03-20]. DOI: 10.1103/PhysRevE.76.026107. ISSN 1539-3755. Dostupné z: <https://doi.org/10.1103/PhysRevE.76.026107>

- [12] Holme, Petter, Sung MIN PARK, Beom Jun KIM a Christofer R. EDLING. Korean university life in a network perspective: Dynamics of a large affiliation network. *Physica A: Statistical Mechanics and its Applications* [online]. 2007, **373**, 821–830 [cit. 2017-02-14]. DOI: 10.1016/j.physa.2006.04.066. ISSN 03784371. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0378437106004882>
- [13] Kovář, Petr. *Úvod do teorie grafů* [online]. FEI VŠB - TUO, 2016 [cit. 2016-01-04]. Dostupné z: [http://home1.vsb.cz/~kov16/files/uvod\\_do\\_teorie\\_grafu.pdf](http://home1.vsb.cz/~kov16/files/uvod_do_teorie_grafu.pdf)
- [14] Newman, M. E. J. *Networks: an introduction*. New York: Oxford University Press, 2010. ISBN 0199206651.
- [15] Newman, M.E.J. The Structure and Function of Complex Networks. *SIAM Review* [online]. 2003, **45**(2), 167–256 [cit. 2015-10-15]. DOI: 10.1137/S003614450342480. ISSN 1095-7200. Dostupné z: <http://epubs.siam.org/doi/10.1137/S003614450342480>
- [16] Newman, M.E.J a D.J. Watts. Scaling and percolation in the small-world network model. *Physical Review E* [online]. 1999, **60**(6), 7332-7342 [cit. 2016-05-06]. DOI: 10.1103/PhysRevE.60.7332. ISSN 1063-651x. Dostupné z: <https://link.aps.org/doi/10.1103/PhysRevE.60.7332>
- [17] Saramäki, Jari, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, János Kertész. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E* [online]. 2007, **75**(2), [cit. 2017-03-21]. DOI: 10.1103/PhysRevE.75.027105. ISSN 1539-3755. Dostupné z: <https://doi.org/10.1103/PhysRevE.75.027105>
- [18] Tarjan, Robert. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing* [online]. 1972, **1**(2), 146–160 [cit. 2017-02-19]. DOI: 10.1137/0201010. ISSN 1095-7111. Dostupné z: <https://doi.org/10.1137/0201010>.
- [19] Williams, Thomas, Colin Kelley. gnuplot 5.0 [software]. [přístup 2017-02-20]. Dostupné z: <http://www.gnuplot.info>
- [20] Zaki, Mohammed J. a Jr, Wagner Meira. *Data Mining and Analysis: Fundamental Concepts and Algorithms* New York: Cambridge University Press, 2014. ISBN: 0521766338, 9780521766333
- [21] Gephi 0.9.1 [software]. [přístup 2016-10-13]. Dostupné z: <https://gephi.org>

## Přílohy

### A Sekvenční diagram vytvoření a uložení grafu



Obrázek 25: Sekvenční diagram vytvoření a uložení grafu.

## B Příklady reálných sítí

Tabulka 7: Reálné sítě a jejich základní vlastnosti, zdroj [1].

Název sítě	Uzly	Hrany	Orientace hran	$n$	$m$	$\langle k \rangle$
Internet	routery	internetové spojení	Neorientované	192244	609066	2,67
WWW	webové stránky	odkazy	Orientované	325729	1497134	4,60
Rozvodná síť	elektrárny, transformační stanice	vodiče	Neorientované	4941	6594	2,67
Telefonní hovory	klienti	hovory	Orientované	36595	91826	2,51
E-mail	emailové adresy	e-mails	Orientované	57194	103731	1,81
Vědecká spolupráce	vědci	spoluautorství	Neorientované	23133	186936	16,16
Sít herců	herci	kolegové	Neorientované	212250	3054278	28,78
Sít citací	články	citace	Orientované	449673	4707958	10,47
Metabolismus	metabolity	chemické reakce	Orientované	1039	5802	5,84
Interakce proteinů	proteiny	interakce proteinů	Neorientované	2018	2930	2,90

## C Výpisy kódu pro generování náhodných grafů

---

```
private Graph GenerateUndirected()
{
    System.Random rnd = new System.Random();
    Graph g = new Graph(N, Weighted);

    for (int i = 0; i < N; i++)
    {
        for (int j = i + 1; j < N; j++)
        {
            double limit = rnd.NextDouble();
            if (P > limit)
            {
                if (Weighted)
                    g.AddEdge(i, j, WeightInterval.GetRandomWeight(rnd));
                else
                    g.AddEdge(i, j);
            }
        }
    }

    return g;
}
```

---

Výpis 7: Generování náhodného  $G(n, p)$  grafu.

---

```
public override Graph GenerateGraph()
{
    System.Random rnd = new System.Random();
    Graph g;
    if(Directed)
        g = new DirectedGraph(N, Weighted);
    else
        g = new Graph(N, Weighted);

    int edges = M;
    while (edges > 0)
    {
        int v_i = rnd.Next(N);
        int v_j = rnd.Next(N);

        if (v_i == v_j) // ne smycky
            continue;

        if (!g.IsEdge(v_i, v_j)) // ne multihraný
        {
            if (Weighted)
                g.AddEdge(v_i, v_j, WeightInterval.GetRandomWeight(rnd));
            else
                g.AddEdge(v_i, v_j);
            edges--;
        }
    }

    return g;
}
```

---

Výpis 8: Generování náhodného  $G(n, m)$  grafu.

## D Výpis kódu pro generování náhodného grafu s očekávanou stupňovou distribucí

---

```
public override Graph GenerateGraph()
{
    System.Random rnd = new System.Random();
    Graph g = new Graph(N);

    int _2m = DegreeSequence.Sum();
    for (int i = 0; i < N; i++)
    {
        for (int j = i; j < N; j++)
        {
            double limit = rnd.NextDouble();
            double probability;

            if (i != j)
                probability = (double)DegreeSequence[i] * DegreeSequence[j] / (_2m);
            else
                probability = (double)DegreeSequence[i] * DegreeSequence[i] / (2.0 * _2m);

            if (probability >= limit)
                g.AddEdge(i, j);
        }
    }

    return g;
}
```

---

Výpis 9: Generování náhodného grafu s očekávanou stupňovou distribucí.

## E Výpisy kódu pro generování grafů podle modelů malého světa

---

```
protected void CreateGrid(Graph g)
{
    int iterace = 1;
    while (iterace <= K)
    {
        for (int i = 0; i < N; i++)
        {
            g.AddEdge(GetVertex(i), GetVertex(i + iterace));
        }
        iterace++;
    }
}
```

---

Výpis 10: Vytvoření mřížky pro modely malého světa.

---

```
public override Graph GenerateGraph()
{
    System.Random rnd = new System.Random();
    Graph g = new Graph(N);

    CreateGrid(g);
    foreach (ProtectedEdge edge in VirtualGrid())
    {
        double limit = rnd.NextDouble();
        if (P > limit)
        {
            int v_k;
            do
            {
                v_k = rnd.Next(N);
            } while (edge.v_i == v_k || g.IsEdge(edge.v_i, v_k));

            g.RemoveEdge(edge.v_i, edge.v_j);
            g.AddEdge(edge.v_i, v_k);
        }
    }

    return g;
}
```

---

Výpis 11: Generování WS modelu.

---

```
public override Graph GenerateGraph()
{
    System.Random rnd = new System.Random();
    Graph g = new Graph(N);

    CreateGrid(g);
    for (int i = 0; i < N * K; i++)
    {
        double limit = rnd.NextDouble();
        if (P > limit)
        {
            int v_i = rnd.Next(N);
            int v_j = rnd.Next(N);
            g.AddEdge(v_i, v_j);
        }
    }

    return g;
}
```

---

Výpis 12: Generování WN modelu.



## F Power-law a CCDF

V této kapitole bude ukázán vztah rozdělení pravděpodobnosti spojité náhodné veličiny řídicí se mocninným zákonem (power-law) k doplňkové kumulativní distribuční funkci (CCDF) tohoto rozdělení.

Podle vztahu 16 je hustota pravděpodobnosti spojité náhodné veličiny řídicí se mocninným zákonem dána jako

$$p(k) = Ck^{-\alpha}, \quad (56)$$

kde  $C$  je normalizační konstanta. Protože  $p(k)$  je hustota pravděpodobnosti spojité náhodné veličiny, bude platit

$$\int_{k_{min}}^{\infty} p(k)dk = 1. \quad (57)$$

Dosazením můžeme určit normalizační konstantu  $C$  následujícím způsobem (za splnění podmínky  $\alpha > 1$ ):

$$C = \frac{1}{\int_{k_{min}}^{\infty} k^{-\alpha} dk} = \frac{1}{\left[ \frac{k^{-\alpha+1}}{-\alpha+1} \right]_{k_{min}}^{\infty}} = \frac{1}{0 - \left( \frac{k_{min}^{-\alpha+1}}{-\alpha+1} \right)} = \frac{\alpha - 1}{k_{min}^{-\alpha+1}}. \quad (58)$$

Hustotu pravděpodobnosti spojité náhodné veličiny řídicí se mocninným zákonem pak můžeme zapsat jako

$$p(k) = \frac{\alpha - 1}{k_{min}^{-\alpha+1}} k^{-\alpha}, \quad (59)$$

čímž jsme mimo jiné dostali vztah 17.

CCDF je funkce  $\bar{F}(x)$  udávající pravděpodobnost, že náhodná veličina  $X$  je větší než zadaná hodnota  $x$ . V našem případě pak bude platit

$$\begin{aligned} \bar{F}(k) &= P[X > k] = \int_k^{\infty} p(x)dx = C \int_k^{\infty} x^{-\alpha} dx = C \cdot \frac{k^{-\alpha+1}}{\alpha - 1} \\ &= \frac{\alpha - 1}{k_{min}^{-\alpha+1}} \cdot \frac{k^{-\alpha+1}}{\alpha - 1} = \left( \frac{k}{k_{min}} \right)^{-\alpha+1} = \left( \frac{k}{k_{min}} \right)^{-(\alpha-1)} \sim k^{-(\alpha-1)}. \end{aligned} \quad (60)$$

Z toho plyne, že pokud se distribuce stupňů vrcholů řídí mocninným zákonem, tak její CCDF se bude také řídit mocninným zákonem. Pokud následně určíme exponent mocninného zákona této CCDF, který označíme  $\beta$ , pak hledaný exponent  $\alpha = \beta + 1$ .

## G Přílohy na přiloženém DVD

- Tento dokument v elektronické podobě
- Aplikace *NetworkGenerator* pro generování sítí včetně zdrojových kódů
- Jednoduchá aplikace pro převod vyexportované distribuce stupňů do formátu s použitím logaritmického „binningu“
- Jednoduchá aplikace pro převod vyexportované distribuce stupňů do formátu s použitím CCDF