

Zadání úlohy 2paths - DataDriven optimalizace WiFi pokrytí pomocí analýzy cest v grafu

Cíl úlohy

Hlavní cíl: Osvojit si aplikaci grafových algoritmů na reálná síťová data pomocí knihovny NetworkX pro analýzu a optimalizaci WiFi infrastruktury.

Co se naučíte:

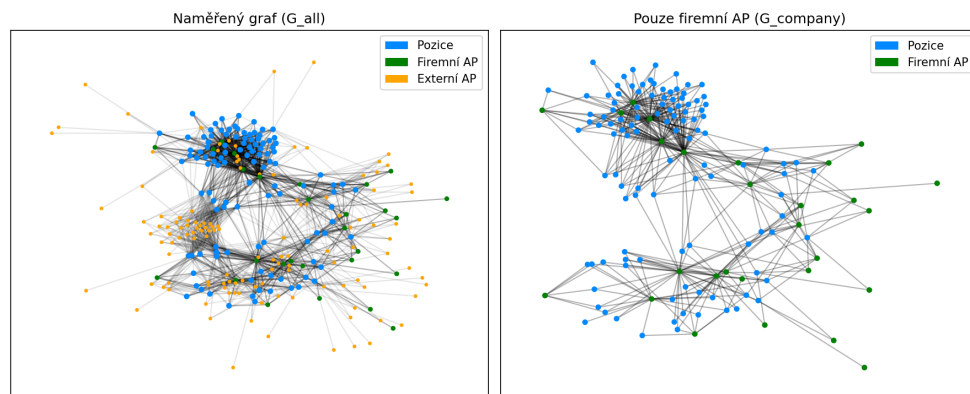
- Modelovat reálné síťové systémy pomocí bipartitních grafů a jejich projekcí
- Používat NetworkX pro analýzu komplexních sítí (centralita, clustering, shortest paths)
- Klasifikovat a filtrovat data podle vzorů pro identifikaci relevantních komponent sítě
- Vizualizovat a interpretovat výsledky grafových analýz pro praktická doporučení

Motivace

Firma se rozhodla vylepšit pokrytí své WiFi infrastruktury na základě data-driven přístupu. Místo intuitivních odhadů chce využít grafové algoritmy k identifikaci redundantních access pointů, optimalizaci tras signálu a nalezení kritických míst vyžadujících posílení. Jako konzultant pro zlepšení WiFi pokrytí máte bipartitní graf $AP \leftrightarrow \text{pozice}$ s vahami podle RSSI.

Manažer chce vědět:

- Kde přidat nové AP?
- Které AP jsou kritické?
- Jak dobře funguje roaming?



WiFi Infrastructure



Zadání

Máte k dispozici CSV soubor `measurement.csv` obsahující 1,758 měření z WiFi sítě. Každé měření obsahuje informace o pozici (MGID), access pointu (BSSID), síle signálu (RSSI) a dalších parametrech. **Analyzujte tato data pomocí NetworkX knihovny** a vytvořte komplexní doporučení pro optimalizaci infrastruktury založené na analýze cest v grafu.



Příprava (1.1 bodu)

1.  **Data loading** (0.2 bodu)
 - načtěte data z CSV
 - zjistěte kolik je záznamů
 - zjistěte kolik je unikátních hodnot: MGID, SSID, BSSID
2.  **Klasifikujte firemní AP** (0.2 bodu)
 - vytvořte graf (sít') $SSID \leftrightarrow MGID(\text{pozice})$
 - vyberte SSID s největším stupněm (je k dispozici na nejvíce pozicích)
3.  **Grafy a projekce** (0.7 bodu)
 - vytvořte graf (AP) $BSSID \leftrightarrow \text{pozice}$
 - k AP vrcholům si uložte příznak “firemní”
 - ke každé hraně si uložte “RSSI” (kvalita signálu)
 - 🤖 vypište počet vrcholů, počet vrcholů v každé paritě, počet hran (bude často, doporučuji definovat funkci)
 - vytvořte podgraf firemní-AP $\leftrightarrow \text{pozice}$
 - 🤖 vypište počet vrcholů, počet vrcholů v každé paritě, počet hran
 - vytvořte projekci na pozice: firemní-AP $\leftrightarrow \text{pozice} \Rightarrow G(\text{pozice})$
 - použijte networkx funkci
 - každé hraně (p1,p2) nastavte:
 - RSSI: $\max(\min(p1-ap[RSSI], p2-ap[RSSI]), ap: sdílené\ ap)$
 - weight: Převod RSSI na váhu pro shortest path (Lepší RSSI = menší váha = “kratší” cesta)
 - shared_aps: počet sdílených ap
 - 🤖 vypište počet vrcholů, počet hran
 - vytvořte projekci na AP: firemní-AP $\leftrightarrow \text{pozice} \Rightarrow G(AP)$
 - použijte networkx funkci
 - každé hraně (p1,p2) nastavte:
 - weight: počet sdílených pozic (💡 weighted_projection)
 - 🤖 vypište počet vrcholů, počet hran



Analýza (1.9 bodu)

1. **Analýza grafu pozic $G(\text{📍})$** (1.3 bodu)
 - Kolikrát se v průměru přepne AP, když se pohybují po budově?
 - vypočítejte průměrnou délku nejkratších cest (bez váhy hrany)
 - vykreslete histogram všech délek
 - interpretujte: “Typicky N přepnutí”
 - Kolikrát se přepne AP nejíce?
 - vypočítejte průměr (diameter) grafu
 - Které pozice jsou nejhůře pokryté/dosažitelné?
 - vypočítejte excentricitu
 - vrcholy s maximální hodnotou označte jako “Mrtvé konce sítě”
 - Kde pokrytí posílit a naopak kde je redundance?

- vypočtete centrum a periferii grafu
-  výsledek vizualizujte obrázkem grafu s barevným rozlišením vrcholů
- Jaké jsou nejlepší a nejhorší roamingové cesty?
 - vypočtete vážené nejkratší cesty (s váhou hrany ~ nepřímo úměrnou RSSI)
 - vyextrahujte RSSI podél cesty a spočtete průměrnou RSSI
 - z 10 nejdelších nejkratších cest určete (vrcholy jsou daleko, ale cesta je nejkratší)
 -  cestu s nejlepším pokrytím
 -  cestu s nejhorším pokrytím
 -  výsledek vizualizujte obrázkem grafu s barevným rozlišením cest  
- 2. **Analýza redundance z grafu AP G()** (0.6)
 - pro každé AP vypočtete podíl cest procházejících uzlem (betweenness centrality)
 - seřaďte AP sestupně 
 - vytvářejte podgrafy 
 - odeberete více a více AP z 
 - zastavte se když podgraf rozpadne na více komponent 
 - Vypiště které AP lze “beztržně” vypnout.
 - tj. než se podgraf rozpadne 

Bodování (celkem 3 body)

Úkol	Body	Popis
 Data loading	0.20	Načtení a základní analýza dat
– Načtení CSV	0.05	Použití pandas pro načtení measurement.csv
– Zjištění počtu záznamů	0.05	Celkový počet řádků v datasetu
– Počet unikátních hodnot	0.10	Unikátní hodnoty pro MGID, SSID, BSSID
 Klasifikace firemních AP	0.20	Identifikace firemní Wi-Fi sítě
– Vytvoření grafu SSID↔MGID	0.10	Bipartitní graf síťových jmen a pozic
– Výběr SSID s max stupněm	0.10	Nalezení nejrozšířenější sítě (firemní AP)
 Grafy a projekce	0.70	Tvorba grafových reprezentací
– Graf BSSID↔pozice s atributy	0.20	Bipartitní graf s příznakem “firemní” a RSSI na hranách
– Podgraf firemní-AP↔pozice	0.10	Filtrování pouze firemních přístupových bodů
– Projekce na pozice G()	0.25	Projekce s atributy: RSSI, weight, shared_aps
– Projekce na AP G()	0.15	Projekce s atributem weight (sdílené pozice)
 Analýza grafu pozic	1.30	Detailní analýza pokrytí budovy
– Průměrná délka cest + histogram	0.25	Typický počet přepnutí AP při pohybu
– Diameter grafu	0.10	Maximální počet přepnutí AP
– Excentricita + mrtvé konce	0.20	Identifikace nejhůře dosažitelných pozic
– Centrum a periferie + vizualizace	0.40	Graf s barevným rozlišením podle důležitosti pozic

Úkol	Body	Popis
– Nejlepší/nejhorší roamingové cesty	0.35	Analýza kvality signálu podél 10 nejdelších cest + vizualizace
 Analýza redundance	0.60	Optimalizace počtu přístupových bodů
– Betweenness centrality	0.10	Důležitost každého AP pro konektivitu
– Seřazení AP sestupně	0.05	Pořadí AP podle centrality
– Postupné odebírání + detekce rozpadu	0.30	Simulace vypínání AP až do ztráty konektivity
– Identifikace “beztrestných” AP	0.15	Seznam AP, které lze vypnout bez dopadu
CELKEM	3.00	

Souhrn podle náročnosti

Náročnost	Body	Počet úkolů
 Jednoduché (≤ 0.10)	0.70	7 úkolů
 Střední ($0.15-0.25$)	1.05	5 úkolů
 Náročné (≥ 0.30)	1.25	3 úkoly

Checklist před odevzdáním

Data loading

- ☐ Načíst CSV soubor se správným kódováním (UTF-8/cp1250)
- Vypsát základní statistiky:
 - ☐ počet AP (unikátní BSSID),
 - ☐ počet pozic (unikátní MGID/Timestamp),
 - ☐ počet měření (celkový počet záznamů)

Klasifikace firemních AP

- ☐ Vytvořit bipartitní graf SSID $\leftrightarrow$ MGID (pozice)
 - ☐ nastavit bipartite atribut uzlům
 - ☐ vypsát počet vrcholů v každé partitě
- ☐ Identifikovat firemní SSID
 - ☐ spočítat stupeň každého SSID uzlu
 - ☐ vybrat SSID s maximálním stupněm
 - ☐ vypsát název firemního SSID

Grafy a projekce

- ☐ Vytvořit bipartitní graf BSSID $\leftrightarrow$ pozice
 - ☐ přidat všechny AP (BSSID) a pozice (MGID/Timestamp)
 - ☐ nastavit atribut “firemní” (boolean) pro AP uzly
 - ☐ nastavit atribut “RSSI” pro hrany
 - ☐ vypsát: počet vrcholů celkem, počet AP, počet pozic, počet hran
- ☐ Vytvořit podgraf firemní-AP $\leftrightarrow$ pozice

- ☐ filtrovat pouze uzly s atributem “firemní” = True
- ☐ vypsát: počet vrcholů celkem, počet firemních AP, počet pozic, počet hran
- ☐ Vytvořit projekci na pozice G()
 - ☐ použít NetworkX projekci (weighted nebo bipartite projection)
 - ☐ pro každou hranu (p1, p2) nastavit:
 - ☐ RSSI: maximum z minima RSSI přes sdílené AP
 - ☐ weight: převod RSSI na váhu (lepší RSSI = menší váha)
 - ☐ shared_aps: počet sdílených firemních AP
 - ☐ vypsát: počet vrcholů, počet hran
- ☐ Vytvořit projekci na AP G()
 - ☐ použít NetworkX weighted projection
 - ☐ pro každou hranu (ap1, ap2) nastavit:
 - ☐ weight: počet sdílených pozic
 - ☐ vypsát: počet vrcholů, počet hran

Analýza grafu pozic G()

- ☐ **Průměrný počet přepnutí AP**
 - ☐ spočítat všechny nejkratší cesty (bez váhy)
 - ☐ vypočítat průměrnou délku
 - ☐ vytvořit histogram délek cest
 - ☐ interpretovat: “Typicky X přepnutí AP při pohybu”
- ☐ **Maximální počet přepnutí**
 - ☐ vypočítat diameter grafu
 - ☐ vypsát hodnotu
- ☐ **Mrtvé konce sítě**
 - ☐ vypočítat excentricitu pro všechny uzly
 - ☐ najít uzly s maximální excentricitou
 - ☐ vypsát seznam “mrtvých konců”
- ☐ **Centrum a periferie + vizualizace**
 - ☐ vypočítat centrum grafu (uzly s min excentricitou)
 - ☐ vypočítat periferii grafu (uzly s max excentricitou)
 - ☐ vytvořit layout grafu (spring/kamada_kawai/...)
 - ☐ vykreslit graf s barevným rozlišením:
 - ☐ centrum (např. zelená)
 - ☐ periferie (např. červená)
 - ☐ ostatní (např. modrá)
 - ☐ přidat legendu
 - ☐ uložit obrázek do souboru
- ☐ **Roamingové cesty - nejlepší a nejhorší**
 - ☐ vypočítat vážené nejkratší cesty (s weight atributem)
 - ☐ seřadit podle délky (počet hopů) sestupně
 - ☐ vybrat 10 nejdelších cest
 - ☐ pro každou cestu:
 - ☐ extrahovat RSSI hodnoty podél cesty
 - ☐ spočítat průměrnou RSSI
 - ☐ identifikovat cestu s nejvyšší průměrnou RSSI ( nejlepší)
 - ☐ identifikovat cestu s nejnižší průměrnou RSSI ( nejhorší)
 - ☐ vizualizace:
 - ☐ vykreslit graf všech pozic
 - ☐ zvýraznit nejlepší cestu (zelená/silná čára)
 - ☐ zvýraznit nejhorší cestu (červená/silná čára)
 - ☐ přidat legendu s průměrným RSSI
 - ☐ uložit obrázek

Analýza redundance G()

- ☐ **Betweenness centrality**
 - ☐ vypočítat betweenness pro všechny AP uzly
 - ☐ seřadit AP sestupně podle betweenness
 - ☐ vypsát top 10 nejdůležitějších AP
- ☐ **Simulace vypínání AP**
 - ☐ vytvořit kopii grafu G()
 - ☐ iterativně odebírat AP v pořadí (od nejvyšší betweenness):
 - ☐ odstranit uzel z grafu
 - ☐ zkontrolovat konektivitu (is_connected nebo počet komponent)
 - ☐ zastavit pokud graf není souvislý (rozpad na komponenty)
 - ☐ zaznamenat seznam odebraných AP před rozpadem
- ☐ **Identifikace beztestných AP**
 - ☐ vypsát seznam AP, které lze vypnout bez ztráty konektivity
 - ☐ vypočítat procento redundantních AP
 - ☐ interpretovat výsledek

Dokumentace a odevzdání

- ☐ Zpráva obsahuje:
 - ☐ jméno autora
 - ☐ popis postupu řešení (může být ipynb)
 - ☐ všechny vypočtené hodnoty a statistiky (může být příloha/ipynb)
 - ☐ všechny grafy a vizualizace s popisky (také přílohy/ipynb)
 - ☐ interpretaci výsledků pro každou sekci (může být příloha/ipynb)
 - ☐ závěr
 - ☐ Zdrojové kódy:
 - ☐ jsou přiloženy k odevzdání
 - ☐ jsou komentované
 - ☐ obsahují README s instrukcemi pro spuštění
 - ☐ Formát odevzdání:
 - ☐ zpráva v PDF
 - ☐ zdrojové kódy v ZIP/TAR
 - ☐ odevzdáno do systému Upload/Brute
 - ☐ deadline dodržen
-

AI tipy pro řešení

Data loading a základy

 **AI tip:** “Jak v pythonu načtu CSV soubor jako list slovníků s utf-8 kódováním, separátor ‘;’?”

 **AI tip:** “Proč tam mám \ufffd ({' \ufffdID': '887', })? Normálně ho v CSV nevidím? Jak ho odstranit při načítání?”

 **AI tip:** “Mám záznamy jako [{'ID': '887', 'MGID': '53', 'BSSID': '74:ea:3a:f4:c6:82', 'SSID': 'TERE', 'RSSI': '-79', ...}]. Jak v pythonu získám počet unikátních BSSID, MGID a celkový počet záznamů?”

Vytváření grafů

 **AI tip:** “Jak v NetworkX vytvořím bipartitní graf ze seznamu hran? Jak nastavím atribut ‘bipartite’ pro dva typy uzlů?”

 **AI tip:** “Jak v NetworkX přidám hrany s atributy? Chci ke každé hraně přidat RSSI hodnotu a boolean ‘firemní’.”

 **AI tip:** “Jak v NetworkX vytvořím projekci bipartitního grafu? Potřebuji weighted projection kde váha = počet sdílených sousedů.”

 **AI tip:** “Jak v NetworkX přistupuji k atributům hran? Potřebuji přechíst RSSI hodnotu pro hranu mezi uzly u a v.”

Převody a výpočty

 **AI tip:** “Jak převést RSSI (-90 až -30 dBm) na váhu pro shortest path? Lepší signál (vyšší RSSI) = menší váha. Jaký vzorec použít?”

 **AI tip:** “Jak v NetworkX nastavím více atributů jedné hraně najednou? Chci nastavit ‘RSSI’, ‘weight’ a ‘shared_aps’.”

Analýza a vizualizace

 **AI tip:** “Jak v NetworkX vytvořím histogram délek všech nejkratších cest mezi všemi páry uzlů?”

 **AI tip:** “Jak vytvořím layout pro bipartitní graf v NetworkX s pozicemi vlevo a AP vpravo? Jak přidám barvy podle typu uzlu?”

 **AI tip:** “Jak v NetworkX najdu cestu mezi dvěma uzly a extrahuji hodnoty atributu ‘RSSI’ pro všechny hrany podél této cesty?”

 **AI tip:** “Jak zvýrazním dvě konkrétní cesty v NetworkX grafu různými barvami? Chci jednu zelenou (best) a jednu červenou (worst) se silnějšími hranami.”

 **AI tip:** “Jak vytvořím matplotlib legendu pro NetworkX graf s vysvětlením: centrum (zelená), periferie (červená), ostatní (modrá)?”

Pokročilá analýza

 **AI tip:** “Jak v NetworkX iterativně odebíráím uzly a kontroluji, zda je graf stále souvislý (connected)? Chci zastavit když se rozpadne na komponenty.”

 **AI tip:** “Jak v NetworkX zjistím, které uzly jsou v centru grafu (minimální excentricita) a které na periferii (maximální excentricita)?”

Bonus

 **Bonus tip:** Při práci s NetworkX dokumentací se AI ptěj: “Jaká NetworkX funkce počítá [betweenness centrality / diameter / eccentricity]? Ukaž mi příklad použití.”

Užitečné knihovny

NetworkX - <https://networkx.org/>

```
import networkx as nx

# Užitečné funkce:
nx.Graph()                # Vytvoření grafu
nx.add_edge(u, v, weight=w) # Přidání hrany s váhou
nx.shortest_path(G, source, target) # Nejkratší cesta
nx.betweenness_centrality(G) # Míra centrality
nx.eccentricity(G)         # Excentricita uzlů
nx.all_simple_paths(G, s, t, cutoff=k) # Všechny jednoduché cesty
nx.bipartite.projected_graph(G, nodes) # Projekce bipartitního grafu
```

Pandas & Matplotlib

```
import pandas as pd
import matplotlib.pyplot as plt

# Načtení s korektním kódováním
df = pd.read_csv('measurement.csv')
plt.figure(figsize=(12, 8))
plt.hist(data, bins=30, alpha=0.7, label='dataset')
```

Toto cvičení vám poskytne praktické zkušenosti s aplikací grafových algoritmů na reálná data WiFi sítě. Naučíte se analyzovat konektivitu, identifikovat kritická místa a optimalizovat síťovou infrastrukturu pomocí NetworkX knihovny.