

Tester SQL úloh na báze evolučných algoritmov

NIKOLA KULÍKOVÁ

ŠKOLITEĽ: ING. ALEXANDER ŠIMKO, PHD.

Problematika

- DB tester na predmete Databázy(1)
- Kontrola študentového riešenia - SQL dopytu
- Vytvorenie 2 databáz z študentového dopytu a uloženého
- Problém pri cudzích kľúčoch

Súčasnέ riešenie

- Namapovanie riadkov na seba, ak taký už nie je tak na null
- Vytvorenie bipartitného grafu
- Vyber najlepšieho mapovania cez Hungarian algorithm
 - Najmenšie ohodnotenie
- Ohodnotenie vzdialenosti riadkov = rozdiel znakov vo všetkých stĺpcoch
- Ak je jeden riadok null tak vráti počet stĺpcov
- Mapovanie obojsmerne -> hodnota hrany je $2 * \text{rozdiel znakov vo všetkých stĺpcoch}$
- Pri rôznych idčkách sa nevie vysporiadať aj keď sú dáta rovnaké

Ohodnotenia podobnosti

- Na vstupe graf
- Nájdenie podobnosti prvkov -> nie nutne najlepšej
- Na základe susedov a referencie susedov
- Čím väčšia hodnota v matici medzi dvoma vrcholmi tým viac podobne

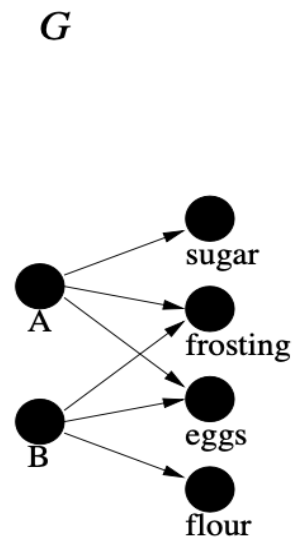
SimRank

1. Inicializuj všetky dvojice na 0.
2. Na diagonále v matici nainicializuj na 1 -> samé na seba
3. Pre každú dvojicu a, b vypočítaj skóre SimRank(a, b):
 1. Ak $a == b$ tak daj 1.
 2. Inak nech $I(a)$ je počet susedov a , $I(b)$ pre vrchol b tak skóre bude:

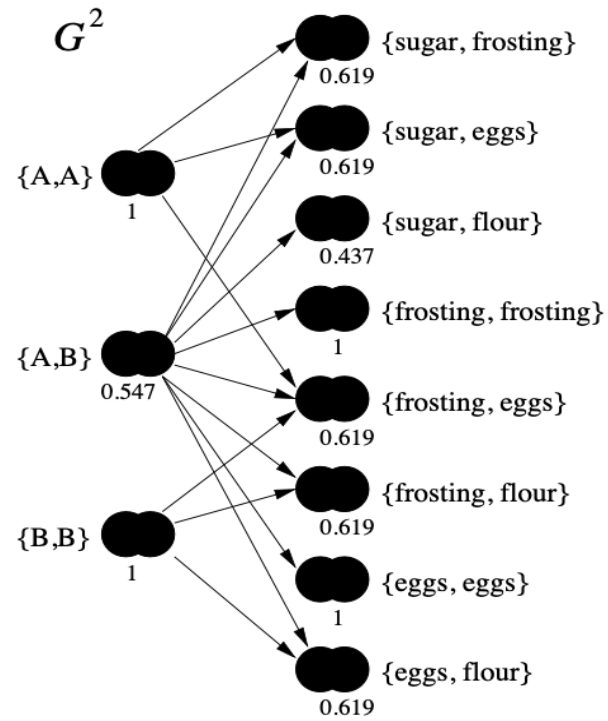
$$s(a, b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$$

3. Opakuj kým neskonverguje

SimRank príklad



(a)



(b)

MatchSim

- Na rozdiel od SimRank vynecháva najlepšie mapovanie -> lepšie výsledky
- Využíva Hungarian algorithm na nájdenie maximálneho mapovania v bipartitnom grafe
- Ako SimRank ale vzorec na priradenie:

$$sim(a, b) = \frac{\widehat{W}(a, b)}{\max(I(a), I(b))}.$$

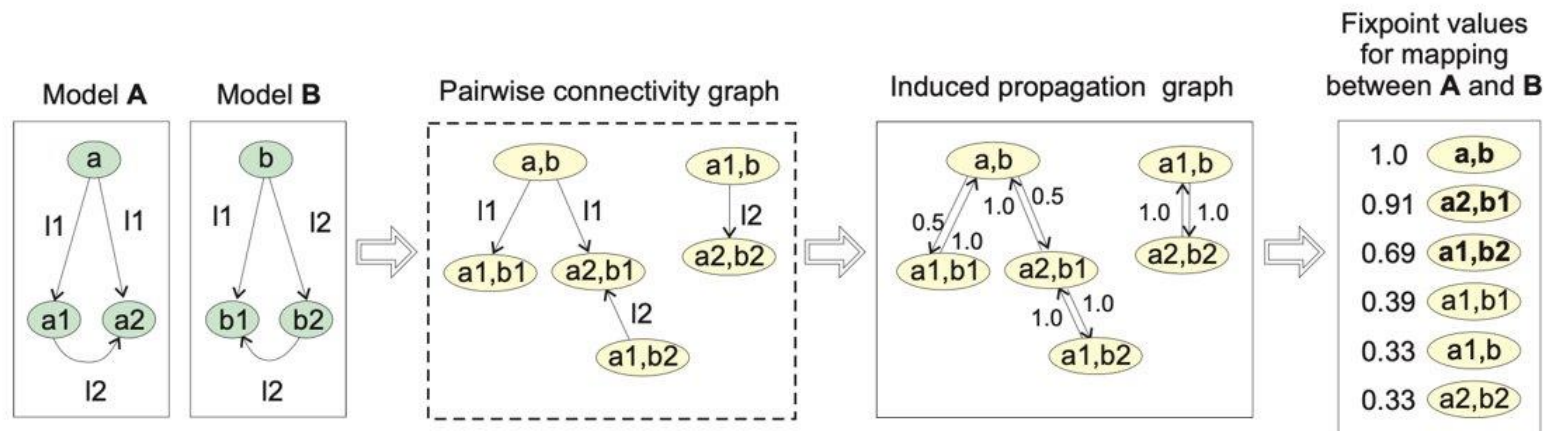
- $W(a, b)$ je váha maximálneho skóre namapovaného -> Hungarian algorithm

Similarity flooding

1. Zmeniť dátový model na ohodnotený graf
2. Pairwise connectivity graph = ak a ma suseda $a1$ a hrana medzi nimi je $l1$, a b ma suseda $b1$ s hranou $l1$, tak vytvorí nový vrchol (a, b) s hranou $l1$ do $(a1, b1)$
-> rekurzívne pospája
3. Propagation graph – pridá aj opačne hrany, zmení hodnoty hrán na pravdepodobnosti podobnosti
4. Vyrátať fixpoint values – v iteráciach podľa vzorca + normalizácia:

$$\sigma^{i+1}(x, y) = \sigma^i(x, y) + \sum_{(a_u, p, x) \in A, (b_u, p, y) \in B} \sigma^i(a_u, b_u) \cdot w((a_u, b_u), (x, y)) + \sum_{(x, p, a_v) \in A, (y, p, b_v) \in B} \sigma^i(a_v, b_v) \cdot w((a_v, b_v), (x, y))$$

Similarity flooding příklad



Evolučný algoritmus

- Počiatočná generácia náhodná
- Jedinec = riešenie
- V danom počte epoch nastáva mutácia a kríženie členov generácie
- Do ďalšej postupuje len polovica najlepších z predchádzajúcej generácie a polka ich detí
- Mutácia = pri náhodnej pravdepodobnosti náhodne zmutovanie génu /informácie
- Kríženie = z 2 členov generácie 1 nový, polka génov od jedného rodiča a polka od druhého

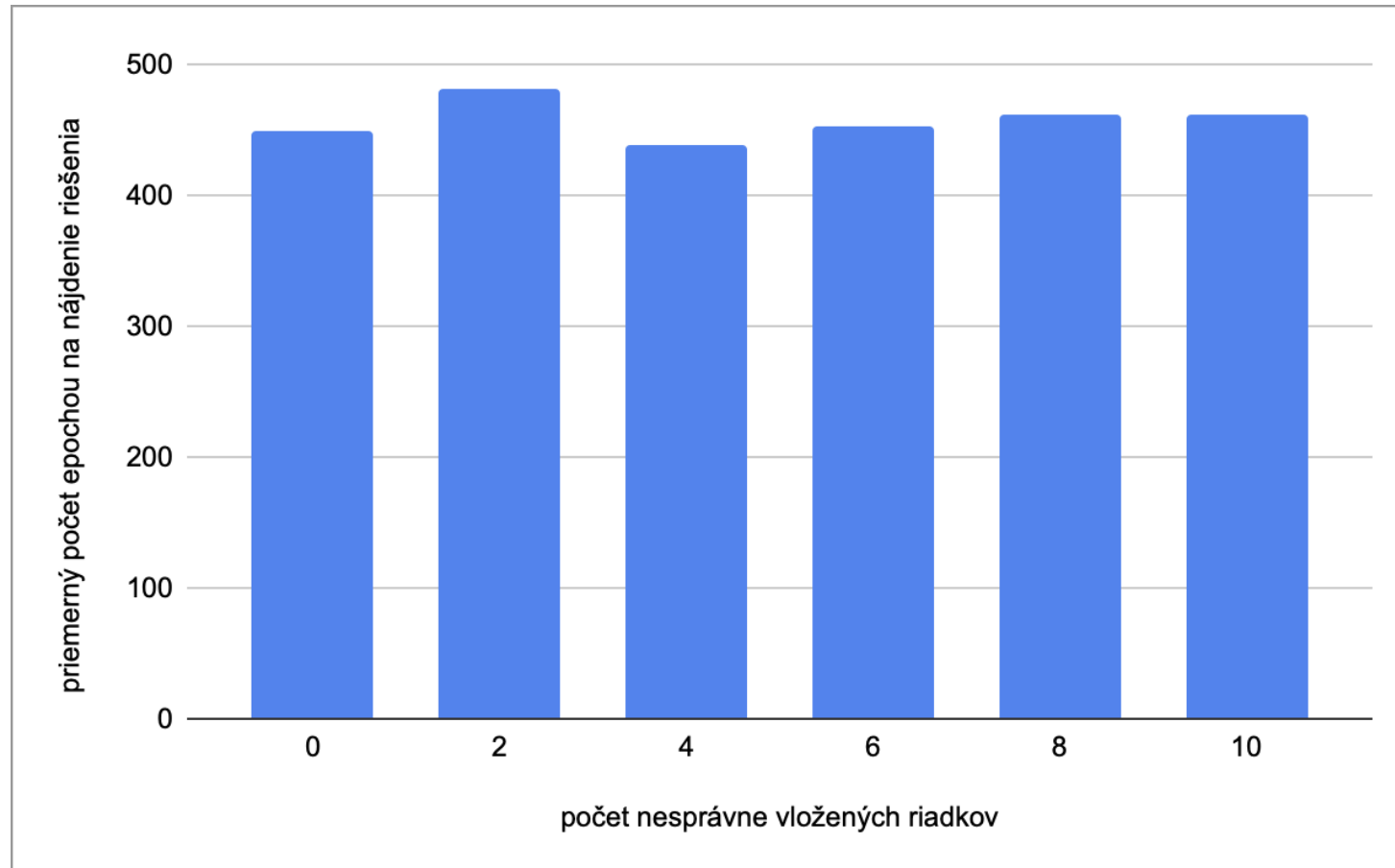
Naimplementované riešenie

- Hľadanie najlepšieho mapovania
- Na začiatku náhodne napamapujeme v každom individuále
- Individuál = zoznam bimáp obsahujúcich dvojice riadkov
- Kríženie = s pravdepodobnosťou 56% pridá z prvého rodiča, ak ešte taký kľúč tam nie je
- Mutovanie = náhodne vybratý druhý riadok a výmena namapovaných riadkov
- Ohodnocovacia funkcia - rozdiel vzdialenosti hodnôt v každom stĺpci a rekurzívne odkazovanie cudzími kľúčami do hĺbky 2

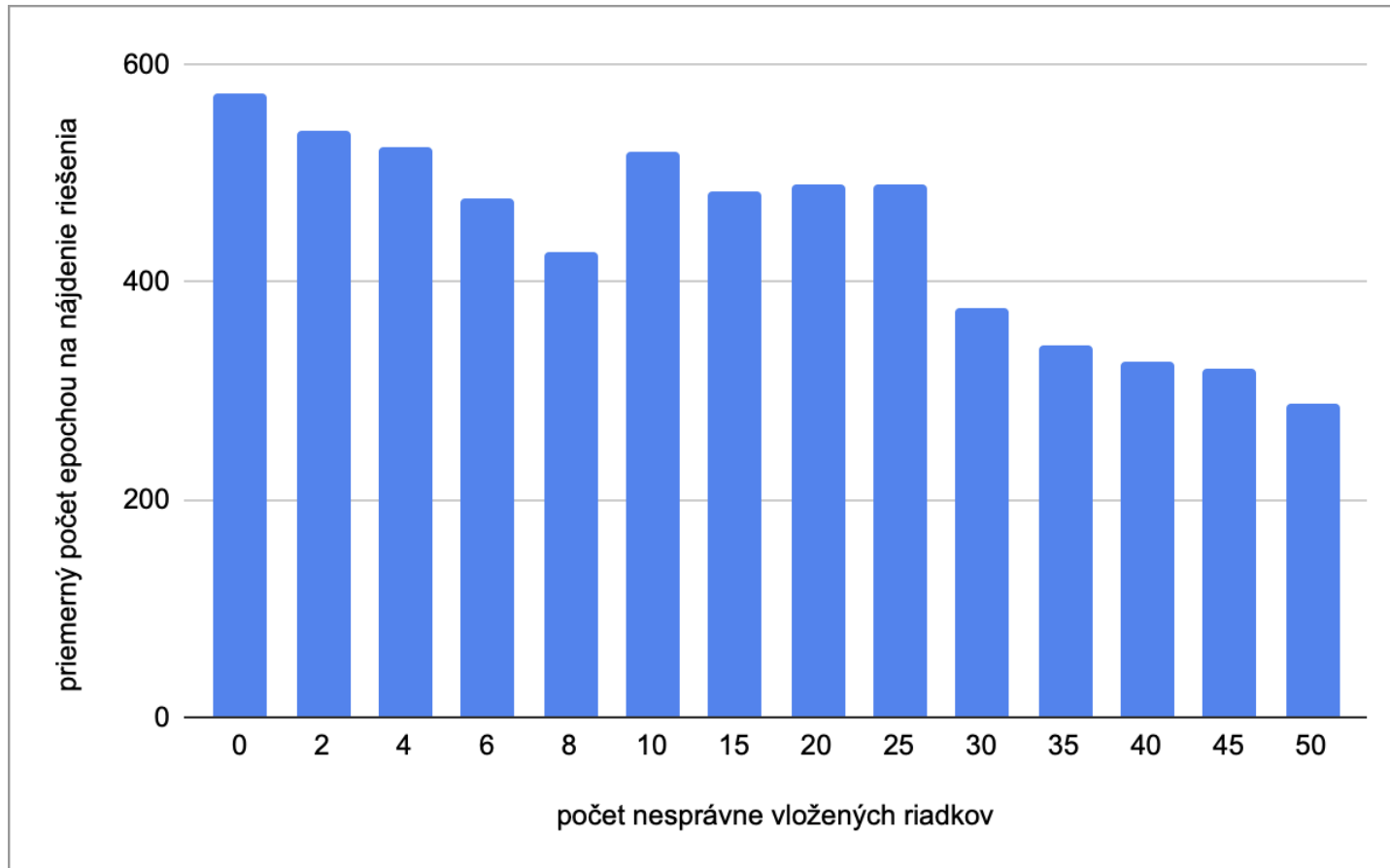
Výsledky

- Dve databázy
- Menšia = 3 tabuľky (zamestnanci, oddelenia, pracovné pozície)
- Najväčšia tabuľka zamestnanci , 91 riadkov
- 2 cudzie kľúče z tabuľky zamestnancov
- Väčšia = 7 tabuliek (zamestnanci, oddelenia, pracovné pozície, krajiny, regióny, oblasti, história pracovných pozícií)
- Viac cudzích kľúčov
- Používaná na cvičeniach z predmetu Databázy(1)

Výsledky - bez zmeny počiatočnej generácie menšia databáza

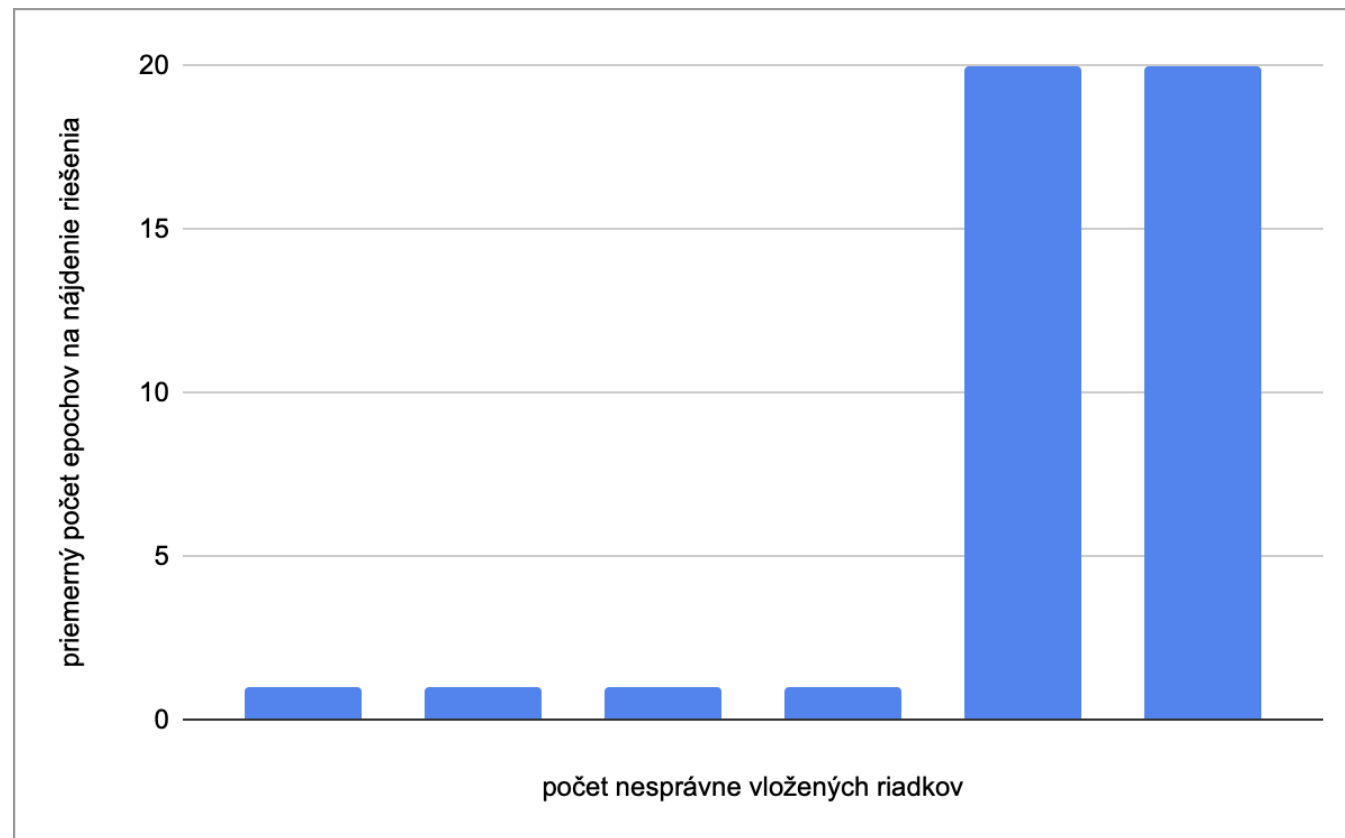


Výsledky - bez zmeny počiatočnej generácie väčšia databáza



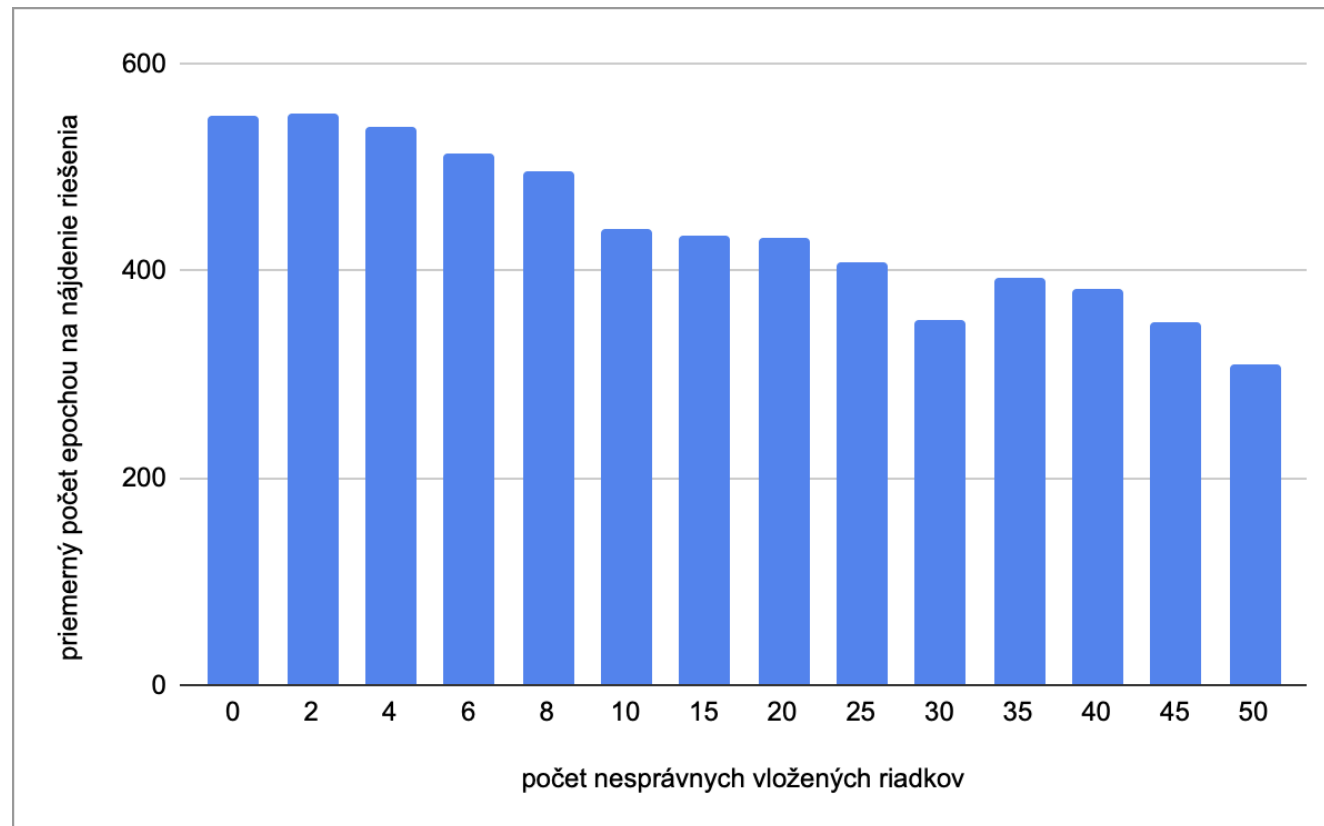
Výsledky - zmena počiatočnej generácie, menšia databáza

- Vytváranie 10 jednotlivcov pomocou maďarského algoritmu



Výsledky - zmena počiatočnej generácie, väčšia databáza

- Vytváranie 10 jedincov pomocou maďarského algoritmu



Zdroje

- SimRank - <http://www.cse.cuhk.edu.hk/~cslui/CMSC5734/simrank.pdf>
- Similarity flooding - <http://ilpubs.stanford.edu:8090/730/1/2002-1.pdf>
- MatchSim - <https://link.springer.com/article/10.1007/s10115-011-0427-z>

Ďakujem za pozornosť
