

UNIVERZA V LJUBLJANI  
FAKULTETA ZA MATEMATIKO IN FIZIKO  
FINANČNA MATEMATIKA

## **Razbitje grafa z odstranjevanjem vozlišč in povezav**

Kratek opis projekta

Avtorici: ADMIRA ABDIĆ, TIA KROFEL  
Ljubljana, november 2021

# 1 Problem

V projektni nalogi si bova podrobneje ogledali problem razbitja grafa z odstranjevanjem vozlišč in povezav. Za vhodni podatek bova vzeli usmerjen povezan graf s podanima vozliščema  $s$  in  $t$ . Naloga projekta je nato pokazati, kako lahko izhodiščni graf razbijemo na več komponent, če smemo odstraniti največ eno vozlišče ter najmanjše možno število povezav. Poleg formulacije problema s pomočjo celoštevilskega programiranja si bova pri reševanju pomagali še z algoritmi za reševanje problema največjega pretoka, ki jih načeloma uporabljamo pri reševanju problemov najmanjšega prereza.

## 2 Teoretično ozadje

Naj bo  $G = (V, E)$  graf, kjer  $V = V(G)$  predstavlja množico vozlišč,  $E = E(G)$  pa množico povezav grafa  $G$ .

Najprej si oglejmo različne načine razbitja posameznega grafa.

Množica  $S \subseteq V(G)$  je *točkovni prerez* na grafu  $G$ , če ima graf  $G - S$  več komponent kot  $G$ . *Povezanost po točkah* grafa  $G$  je enaka minimalni moči množice  $S$ , za katero je  $G - S$  nepovezan (ali ima le eno točko). Graf je  $k$ -povezan po točkah, če je njegova povezanost po točkah vsaj  $k$ .

Množico  $F \subseteq E(G)$  pa imenujemo *razdelitvena množica povezav* na grafu  $G$ , če ima graf  $G - F$  več komponent kot  $G$ . *Povezanost po povezavah* grafa  $G$  je enaka minimalni moči množice  $F$ , za katero je  $G - F$  nepovezan. Graf je  $k$ -povezan po povezavah, če je njegova povezanost po povezavah vsaj  $k$ .

Ker naju bo zanimalo tudi, kako lahko dan problem rešiva s pomočjo algoritmov za reševanje problema največjega pretoka, definiramo še slednjega.

**Problem največjega pretoka:** Naj bo  $G = (V, E)$  usmerjen graf,  $s \in V$  (izvor) in  $t \in V$  (ponor) dani odlikovani vozlišči, število  $c_{ij} \geq 0$  pa naj za vsako povezavo  $(i, j) \in E$  predstavlja prepustnost povezave  $(i, j)$ . Prepustnost lahko na vse pare razširimo na sledeč način:

$$c(i, j) = \begin{cases} c_{ij}, & \text{če } (i, j) \in E, \\ 0, & \text{če } (i, j) \notin E. \end{cases}$$

Urejeno četverko  $(G, s, t, c)$  imenujemo *(pretočno) omrežje*. Na podlagi teh podatkov nato iščemo največji pretok  $f$  v pretočnem omrežju  $(G, s, t, c)$ .

Pri **problemu najmanjšega prereza** imamo prav tako podano pretočno omrežje  $(G, s, t, c)$  in iščemo prerez z najmanjšo prepustnostjo.

Za reševanju danega problema bova implementirali *algoritem Forda in Fulkersona* za reševanje problema največjega pretoka in najmanjšega prereza, ki za vhodni podatek vzame pretočno omrežje  $(G, s, t, c)$ , kot rezultat pa vrne tako največji pretok  $f$  kot tudi najmanjši prerez  $(A, B)$ .

### 3 Načrt dela

Zgoraj opisan problem bova najprej prevedli na celoštevilski linearni program, nato pa za implementacijo algoritma uporabili programski jezik *Python*.

Za posamezen graf bo algoritem najprej preveril, ali je slučajno možno prekiniti povezavo med  $s$  in  $t$  že z odstranitvijo enega samega vozlišča. V tem primeru bo problem rešen in najmanjša množica povezav, ki jih moramo odstraniti, bo prazna.

V nasprotnem primeru bo za vsako vozlišče  $u$  (izvzemši izvor  $s$  in ponor  $t$ ) preveril, kakšen je najmanjši prerez, ki ga vrne algoritem Forda in Fulkersona, če mu podamo pretočno omrežje  $(G - u, s, t, c)$ , ki ga iz grafa  $G - u$  dobimo tako, da za vsako povezavo določimo prepustnost 1, saj nam bo v tem primeru algoritem kot največji pretok (in obenem velikost najmanjšega prereza) vrnil kar število povezav, ki jih bomo morali odstraniti za željeno razbitje.

Nato bo odstranil tisto vozlišče  $u$ , za katerega bo za razbitje grafa  $G - u$  potrebno odstraniti najmanjše število povezav. Le-te bo tudi (zopet na podlagi rezultata Ford in Fulkersonovega algoritma) odstranil in poleg odstranjenega vozlišča kot rezultat vrnil tudi razdelitveno množico povezav na grafu  $G - u$ .

Na koncu pa bova za preizkus opisanega algoritma v Pythonu generirali še nekaj usmerjenih grafov.