

# *Grid-peeling* - kratek opis projekta

Gašper Pust, Mitja Mandić

9. november 2020

## 1 Opis problema

V projektu si bomo podrobneje ogledali konveksne ovojnice  $m \times n$  mreže. Konveksna ovojnica množice je najmanjša konveksna množica, ki vsebuje dano množico. Najlažje si jo predstavljamo tako, kot da bi okoli elementov množice napeli elastiko - kar elastika obkroži, je konveksna ovojnica. Lupljenje konveksnih ovojnic mreže, oziroma angleško *grid* - *peeling* je proces, ko iz mreže iterativno odstranjujemo konveksne ovojnice. S simboli lahko to zapišemo takole:  $P_0 = G_{n,m} = \{1, \dots, n\} \times \{1, \dots, m\}$ . Naj bo  $C_i = \mathcal{CH}(P_{i-1})$  za  $i = 1, \dots$ .  $V_i$  naj bo množica vozlišč  $C_i$  - kot vozlišče razumemo točko, ki je na vogalu mreže (torej za katero bi zataknil elastiko). Naj bo sedaj  $P_i = P_{i-1} \setminus V_i$ . Začnemo torej z  $n \times m$  mrežo in iterativno lupimo konveksne ovojnice, dokler ne odstranimo vseh točk.

V projektni nalogi bova s pomočjo simulacij opazovala v literaturi navedene številke za  $n \times n$  mrežo - teorija napoveduje  $\theta(n^{\frac{4}{3}})$  ovojnic. Za  $n \times m$  mrežo v literaturi ni navedenih podatkov, zanimala naju bo morebitna povezava. Simulacije bova izvedla tudi za točke na neenakomerni mreži.

## 2 Orodja in algoritmi

Za iskanje ovojnic bova v Pythonu implementirala Jarvis-march (gift - wrapping) algoritem ali Graham-scan. Prvi je enostavnejši, vendar nekoliko počasnejši. Jarvisov algoritem na vsakem koraku pregleda vse točke, ki niso v ovojnici in vanjo doda tiste, ki so najbolj levo in najdlje od trenutne točke. Podrobneje ga bomo predstavili v zaključnem poročilu.

Po izvedenem eksperimentalnem delu, bomo rezultate analizirali in jih primerjali z rezultati iz literature. Zanimalo nas bo, kako drugačno je število ovojnic na  $m \times n$  mreži v primerjavi s simetrično.