

# Procenjivanje minimalne površine pravougaonih i konveksnih kontejnera za problem pakovanja konveksnih poligona koristeći translacije

Prikaz naučnog rada u okviru kursa  
Geometrijski algoritmi  
Matematički fakultet

Ivan Ristović

decembar 2018.

## Sažetak

Algoritmi za efikasno pakovanje objekata imaju važne primene. Sam problem ima raznorazne varijante, od kojih su autori izabrali pakovanje konveksnih poligona u kontejner oblika pravougaonika pri čemu su dozvoljene samo translacije poligona. Autori predstavljaju algoritam vremenske složenosti  $O(n \log n)$  za aproksimaciju rešenja datog problema uz dokaz da je dobijena površina najviše 17.45 puta veća od optimalne, što je prvi poznati dokaz da se ovakvi NP-teški problemi uopšte mogu aproksimirati. Takođe, autori daju aproksimaciju rešenja problema pakovanja konveksnih poligona u konveksne kontejnere sa ocenom površine najviše 27 puta većom od optimalne. Istraživanje je najvećim delom izvršeno na međunarodnoj radionici za računarsku geometriju u Barbadosu, dok je odgovarajući rad objavljen 2017. godine u časopisu *Journal of Computational Geometry* pod imenom *Approximating minimum-area rectangular and convex containers for packing convex polygons*, čiji su autori Helmut Alt, Mark de Berg i Christian Knauer.

## Sadržaj

<b>1 Uvod</b>	<b>2</b>
<b>2 Pakovanje u pravougaone kontejnere</b>	<b>2</b>
2.1 Pakovanje poligona iz jedne visinske klase . . . . .	3
2.2 Generisanje i pakovanje mini-kontejnera . . . . .	3
2.3 Detalji implementacije . . . . .	4
<b>3 Pakovanje u konveksne kontejnere</b>	<b>4</b>
<b>Literatura</b>	<b>4</b>

## 1 Uvod

Problem efikasnog pakovanja objekata u kontejnere ima raznolike varijacije i ogromne primene. Dvodimenzionalne verzije problema se javljaju u tzv. *pakovanju u trake* [4], gde se dati skup objekata pakuje u pravougaone trake fiksne širine minimizujući širinu trake. U tri dimenzije, problem se prirodno javlja u minimizaciji prostora prilikom transportovanja objekata.

Oblik objekata je od velikog uticaja na složenost problema. U dve dimenzije možemo razmatrati pravilne poligone, pravougaonike, proste ili konveksne poligone. Pritom, mora se specificovati koje transformacije objekata su dozvoljene - da li ih je dozvoljeno rotirati ili samo translirati.

Već su jednostavne varijante ovog problema NP-teške, recimo pakovanje skupa pravougaonika sa stranicama paralelnim koordinatnim osama koristeći samo translacije, što je pokazano u [3] tako da se polinomijalni algoritmi mogu konstruisati samo ukoliko je broj objekata konstantan [1, 2]. Stoga su konstruisani razni algoritmi za aproksimaciju koristeći svakojahe heuristike, ali mali je broj takvih algoritama za problem nalaženja *kontejnera minimalne površine* za objekte.

S obzirom da je problem pakovanja proizvoljnog broja objekata NP-težak, postavlja se pitanje: *Da li je moguće efikasno aproksimirati rešenje problema pakovanja  $n$  objekata?* Autori rada odgovaraju potvrdno, dajući algoritam za aproksimaciju problema pakovanja skupa  $P$  konveksnih poligona u pravougaoni kontejner minimalne površine (opisan u delu 2). Koristeći ove rezultate moguće je takođe aproksimirati minimalni konveksni kontejner za  $P$  (opisan u delu 3). Oba algoritma imaju vremensku složenost  $O(n \log n)$ .

## 2 Pakovanje u pravougaone kontejnere

**Definicija 2.1.** Neka je  $P = \{p_1, p_2, \dots, p_k\}$  skup  $k$  konveksnih poligona sa ukupno  $n$  temena. Pravougaonik sa stranicama paralelnim koordinatnim osama u koji možemo da spakujemo sve poligone bez njihove prethodne rotacije nazivamo *kontejner* za  $P$ .

Cilj je pronaći kontejner za  $P$  minimalne površine, u daljem tekstu *optimalni kontejner*. Neka je  $b_{opt}$  optimalni kontejner za  $P$  i neka je njegova površina  $OPT$ . Algoritam koji autori predstavljaju nalazi kontejner površine ne veće od  $17.45 \cdot OPT$ .

**Definicija 2.2.** *Visina* (Širina) poligona  $p$ , u oznaci  $height(p)$  ( $width(p)$ ), se definiše kao razlika njegove maksimalne i minimalne  $y$  ( $x$ ) koordinate.

**Definicija 2.3.** *Maksimalna visina* (*širina*) skupa  $P$ , u oznaci  $h_{max}(w_{max})$ , se definiše kao maksimum visina (*širina*) svih poligona iz  $P$ .

Rešenje se zasniva na particionisanju  $P$  u *visinske klase*  $P_0, P_1, \dots$  po visini: Poligoni sa visinom između  $h_{max}$  i  $\alpha h_{max}$  se nalaze u  $P_0$ , poligoni sa visinom između  $\alpha h_{max}$  i  $\alpha^2 h_{max}$  se nalaze u  $P_1$ , itd. Parametar  $\alpha$  se kasnije bira tako da je aproksimativni faktor optimalan. Algoritam se sastoji iz dva velika koraka:

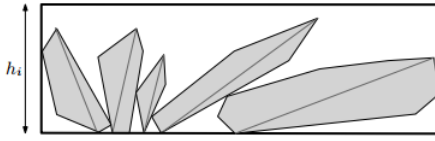
- Spakovati svaku visinsku klasu  $P_i$  zasebno u kontejner  $B_i$  visine  $h_i$ .
- Zameniti svaki neprazni kontejner  $B_i$  kolekcijom *mini-kontejnera* poravnatih sa koordinatnim osama koji nisu previše široki. Spakovati sve mini-kontejnere u jedinstveni kontejner  $B$ .

## 2.1 Pakovanje poligona iz jedne visinske klase

Bez umanjenja opštosti, izaberimo proizvoljnu visinsku klasu  $P_i$ . Poligoni iz te klase imaju visine u opsegu  $(\alpha h_i, h_i]$ . Neka je  $\sigma = [0, \infty) \times [0, h_i]$  polu-beskonačna traka visine  $h_i$ . Poligoni se u  $\sigma$  stavljaju pohlepno.

**Definicija 2.4.** Za poligon  $p$ , neka je  $s(p)$  duž koja spaja najniže i najviše teme poligona  $p$  posmatrajući  $y$ -koordinatu.  $s(p)$  nazivamo *kičmom* poligona  $p$ .

Poligone iz  $P_i$  sortiramo po nagibu njihovih kičmi i postavljamo ih jedan po jedan u  $\sigma$ , pomerajući svaki ulevo sve dok ne dodiruje drugi poligon ili granicu od  $\sigma$  (videti sliku 2.1). Kad se svi poligoni ubace u  $\sigma$ , trivijalno se formiraju granice kontejnera  $B_i$ .



Slika 2.1: Primer pakovanja poligona iz iste visinske klase

**Lema 2.5.** Površina kontejnera  $B_i$  formiranog na gore opisani način zadovoljava:

$$area(B_i) \leq 2/\alpha \cdot \sum_{p \in P_i} area(p) + 2h_i \cdot \max_{p \in P_i} width(p)$$

## 2.2 Generisanje i pakovanje mini-kontejnera

Primenom koraka opisanog u delu 2.1 za sve visinske klase dobijamo kolekciju kontejnera  $B_i$  različitih dužina  $l_i$ . Svaki kontejner sadrži sve poligone iz visinske klase  $P_i$ . Zamenjujemo svaki kontejner  $B_i$  mini-kontejnerima jednakih dužina na sledeći način: Neka je  $w_{max}$  maksimalna dužina poligona iz  $P$ . Prvo, particionišemo  $B_i$  u *kutije* dužine  $cw_{max}$ <sup>1</sup> (vrednost za  $c$  se određuje kasnije) i visine  $h_i$ . Svaki poligon  $p \in P$  se razvrstava u kutiju  $b$  koja sadrži njegovo najlevlje teme<sup>2</sup>. Sada generišemo mini-kontejner za svaku kutiju  $b$  tako što proširujemo  $b$  udesno sve dok njena širina ne postane tačno  $(c+1)w_{max}$ . Dobijamo kolekciju  $\overline{R}_i$  od najviše  $l_i/(cw_{max}) + 1$  mini-kontejnera gde svaki ima istu dužinu. Stoga:

$$\sum_{b \in \overline{R}_i} area(b) \leq (1 + 1/c) \cdot area(B_i) + (c+1)w_{max}h_i$$

Neka je  $\overline{R} = \cup \overline{R}_i$  kolekcija svih mini-kontejnera dobijenih iznad. Nalaženje kontejnera za  $R$  se obavlja trivijalno bez gubitka površine postavljanjem mini-kontejnera jedan na drugi<sup>3</sup>. Tako formiramo kontejner  $B$ .

Može se pokazati da važi:

$$area(B) \leq \underbrace{\left( \left( 1 + \frac{1}{c} \right) \cdot \frac{2 + c\alpha}{\alpha - \alpha^2} \right)}_{f(c, \alpha)} OPT$$

<sup>1</sup>Osim poslednje kutije, koja može da ima dužinu manju od  $w_{max}$ .

<sup>2</sup>Ako najlevlje teme leži na granici između dve kutije, dodeljujemo ga desnoj.

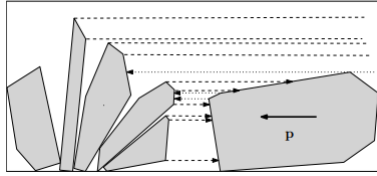
<sup>3</sup>Ovo je moguće jer svi mini-kontejneri imaju istu dužinu.

Kako bi se minimizovao aproksimativni faktor, moraju se naći optimalne vrednosti za  $\alpha$  i  $c$ , tj.  $\alpha \approx 0.407$  i  $c \approx 2.214$ , dajući vrednost aproksimativnog faktora  $f(c, \alpha) \approx 17.449$ .

### 2.3 Detalji implementacije

Kako bi se ispunila obećana vremenska složenost, posmatrajmo složenosti svakog od koraka algoritma. Prvo, particionisanje poligona iz  $P$  u visinske klase se radi u  $O(n \log n)$  vremenu.

Pakovanje poligona opisano u delu 2.1 se može efikasno uraditi održavanjem balansirano binarnog drveta pretrage  $T$ . U svakom čvoru se čuvaju temeena skupa  $P'$  poligona već spakovanih vidljivih sa desna (videti sliku 2.2), uređenih po  $y$ -koordinati. Stoga se ažuriranje  $T$  radi u  $O(\log n)$  vremenu za svako teme, ukupno  $O(n \log n)$ .



Slika 2.2: Struktura podataka za ubacivanje poligona.

## 3 Pakovanje u konveksne kontejnere

Potrebno je pronaći odgovarajuću orijentaciju  $\phi$ , pronaći odgovarajući kontejner  $B$  te orijentacije na osnovu algoritma opisanog u delu 2 i vratiti  $B$  kao rešenje.

Za dati skup poligona  $P$ , biramo  $\phi^*$  minimizujući  $h_{max}(\phi)w_{max}(\phi)$ , gde je  $h_{max}(\phi)$  maksimalni doseg bilo kog poligona u smeru normalnom na  $\phi$  a  $w_{max}(\phi)$  maksimalni doseg u smeru  $\phi$ . Postoje algoritmi za računanje  $\phi^*$  u  $O(n \log n)$  [1].

Autori tvrde i pokazuju da, ukoliko označimo optimalni konveksni kontejner sa  $C_{opt}$ , važi:

$$area(B_{opt}) \leq 2 \cdot area(C_{opt})$$

Slično kao u prethodnom odeljku, nalaze se optimalne vrednosti za  $\alpha$  i  $c$ , naime  $\alpha = 1/3$  i  $c = 2$ , što daje optimizacioni faktor  $f(\alpha, c) = 27$ .

## Literatura

- [1] H. Ahn, H. Alt, S. W. Bae, and D. Park. Bundling three convex polygons to minimize area or perimeter. In *Algorithms and Data Structures - 13th International Symposium, WADS 2013*, August 2013.
- [2] H.-K. Ahn and O. Cheong. *Aligning two convex figures to minimize area or perimeter*. Algorithmica, 2000.
- [3] R. J. Fowler, M. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are np-complete. *Inf. Process. Lett.*, 12(3):133–137, 1981.
- [4] R. Harren, K. Jansen, L. Prädél, and R. van Stee. A  $(5/3 + \epsilon)$ -approximation for strip packing. *Comput. Geom.*, 47(2):248–267, 2014.