

OPTIMIZACIJA SEČENJA DVIDIMENZIONALNIH ELEMENATA PRIMENOM EVOLUTIVNIH ALGORITAMA**OPTIMIZATION OF TWO DIMENSIONAL CUTTING STOCK PROBLEM USING EVOLUTIONARY ALGORITHMS**

Milica Popović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – SOFTVERSKO INŽENJERSTVO I INFORMACIONE TEHNOLOGIJE

Kratak sadržaj – U radu su predstavljena dva evolutivna algoritma za optimizaciju sečenja dvodimenzionalnih elemenata: genetski algoritam i evoluciono programiranje. Implementirane su dve strukture podataka za reprezentaciju jedinki, i različiti mehanizmi za selekciju, ukrštanje i mutaciju jedinki. Implementirana je i varijacija genetskog algoritma koja primenjuje tabu pretragu umesto standardnih mehanizama mutacije. Algoritmi su evaluirani nad problemima za upoređivanje koji zahtevaju sečenje do 2700 elemenata.

Ključne reči: Optimizacija sečenja dvodimenzionalnih elemenata, Genetski algoritam, Evoluciono programiranje, Tabu pretraga

Abstract – This paper presents two evolutionary algorithms for the solution of 2D cutting stock problem: genetic algorithm and evolutionary programming. Two data structures were implemented for the representation of individuals, along with various mechanisms for selection, crossover and mutation. A variation of genetic algorithm that applies tabu search instead of standard mutation mechanisms was also implemented. The algorithms were evaluated on benchmark problems that require cutting up to 2700 elements.

Keywords: 2D Cutting stock problem, Genetic algorithm, Evolutionary programming, Tabu search

1. UVOD

Tema ovog rada je rešavanje problema sečenja dvodimenzionalnih elemenata, sa ciljem da se minimizuje količina utrošenog materijala. Pod dvodimenzionalnim elementima se podrazumevaju elementi koji imaju izražene dve dimenzije, širinu i visinu, i koji su pravougaonog oblika. Prilikom sečenja ovakvih elemenata, svi rezovi moraju biti obavljeni celokupnom širinom, odnosno visinom elementa. Takvi rezovi se u ovom radu nazivaju rezovima od ivice do ivice (eng. *Guillotine cut*). Takođe, dozvoljeni su samo oni rezovi koji su paralelni sa jednom od ivica materijala. U ovom radu su implementirana dva algoritma za rešenja ovog problema, a to su genetski algoritam i evoluciono programiranje.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Zoran Jeličić, red. prof.

2. PREGLED LITERATURE

Zbog kompleksnosti problema optimizacije sečenja dvodimenzionalnih elemenata, i zbog njegove široke primene u industriji, velik broj radova se bavi rešavanjem ovog problema. pristupi za rešavanje problema se mogu ugrubo podeliti na pristupe zasnovane na linearnom programiranju i pristupe zasnovane na heurističkim metodama. Većina ranijih radova rešava ovaj problem primenom tehnike generisanja kolona [1].

Mane pristupa zasnovanih na tehnici generisanja kolona su to što broj kolona eksponencijalno raste sa porastom veličine problema, kao i to što troše mnogo računarskog vremena. Često su dobri za probleme manje i srednje veličine [1]. Kako bi prevazišli ograničenja ovih pristupa, neki od novijih radova su implementirali pristupe zasnovane na evolutivnim algoritmima [2, 3].

3. GENETSKI ALGORITAM

Genetski algoritam je optimizacioni algoritam, koji oponaša proces prirodne evolucije. Faze genetskog algoritma su: inicijalizacija populacije, evaluacija prilagođenosti jedinki, selekcija roditeljskih jedinki, njihovo ukrštanje, mutacija i odabir jedinki naredne generacije. Koraci genetskog algoritma su objašnjeni u nastavku.

3.1. Reprezentacija jedinki

U ovom radu su implementirane dve strukture podataka za reprezentaciju jedinke, a to su izraz sa dva tipa operatora, u postfiksnoj notaciji i permutacija rednih brojeva kojima su označeni traženi elementi.

3.1.1 Jedinka kao izraz u postfiksnoj notaciji

Idea reprezentacije jedinke kao izraza u postfiksnoj notaciji jeste da se predstavi način za međusobno uklapanje elemenata koje je potrebno iseći, tako da oni mogu biti obuhvaćeni elementima standardnih dimenzija. Pri tome se smatra da se dva elementa mogu uklopiti tako što se ređaju horizontalno (operator H) ili vertikalno (operator V) jedan do drugog, i to tako da rezultujući, pravougaoni element koji ih obuhvata bude minimalne površine. Operandi u ovim izrazima su redni brojevi traženih elemenata, kojima se može dodati negativan predznak kako bi se naznačilo to da je element rotiran za 90 stepeni [3].

3.1.2 Jedinka kao permutacija

Reprezentacija jedinke kao permutacije podrazumeva to da se jedinka matematički predstavlja kao permutacija (1):

$$\pi = (x_1, x_2, \dots, x_N) \quad (1)$$

U formuli (1), N predstavlja broj traženih elemenata i takođe važi (2):

$$x_i = \begin{cases} -i, & \text{element } i \text{ je rotiran} \\ i, & \text{element } i \text{ nije rotiran} \end{cases} \quad (2)$$

To znači da je jedinka reprezentovana kao sekvenca prema kojoj traženi elementi mogu biti upakovani u kontejner fiksne širine i beskonačne visine, po heuristici *dole ulevo* (eng. *Bottom Left*) [2].

Prilikom pakovanja svakog od traženih elemenata u kontejner, potrebno je odabrati pravougaoni deo slobodnog prostora u koji će posmatrani element biti upakovan. Potom se posmatrani element postavlja u donji levi ugao pravougaonog dela slobodnog prostora. Nakon pakovanja elementa, preostali slobodni prostor se može podeliti na dve nove, disjunktne pravougaone površine, i to na dva načina: po horizontalnoj osi i po vertikalnoj osi. Stoga je neophodno odrediti heuristike po kojima će se vršiti odabir dela slobodnog prostora za pakovanje i podela preostalog dela slobodnog prostora nakon pakovanja. U ovom radu je kao heuristika za odabir dela slobodnog prostora za pakovanje primenjena heuristika koja minimizuje dužinu preostale kraće stranice pravougaonog dela slobodnog prostora [4]. S druge strane, kao heuristika za podelu preostalog dela slobodnog prostora je implementirana heuristika koja deli prostor po kraćoj osi, nezavisno od dimenzija preostalog slobodnog prostora [4].

3.2. Generisanje početne populacije

Da bi mogao započeti proces pretrage prostora rešenja, potrebno je generisati inicijalnu populaciju. Za to je najpre potrebno odrediti njenu veličinu. Veličina populacije može značajno da utiče na uspešnost i performanse algoritma. Ukoliko je populacija premala, algoritam ima tendenciju da prerano konvergira do lokalnog optimuma. S druge strane, ako je populacija prevelika, dolazi do uvećanja računarske složenosti algoritma. U ovom radu je veličina populacije postavljena na 800, za algoritme koji ne implementiraju tabu pretragu, a za algoritme koji je implementiraju na 200.

Jedinke koje će biti deo inicijalne populacije se generišu nasumično. Ukoliko je jedinka reprezentovana kao izraz, potrebno je voditi računa o tome na kojim pozicijama se smeju nalaziti operatori, kako bi izraz bio validan. Takođe, u generisanom izrazu broj operatora mora biti za jedan manji od broja operanada.

3.3. Evaluacija

Za svaku jedinku u populaciji je potrebno odrediti koliko je ona kvalitetno rešenje problema. U ovom radu, cilj je minimizovati broj upotrebljenih elemenata standardnih dimenzija, i količinu otpada, a pri tome zadovoljiti potrebe naručioca. Pojam količine otpada se u ovom kontekstu odnosi na neiskorišćeni deo minimalne pravougaone površine koja obuhvata tražene elemente, u sklopu svakog upotrebljenog elementa standardnih dimenzija.

Funkcija prilagođenosti je definisana kao funkcija troška, odnosno cilj je da njena vrednost bude minimalna. Funkcija troška je zadata formulom (3):

$$|S| + \frac{W \cdot H^* - \sum_{i=1}^N w_i \cdot h_i}{W \cdot H^*} + \max(0, W^* - W) + \frac{|\{s \in S | h_s > H\}|}{|S|} \cdot 100 \quad (3)$$

U formuli (3), S predstavlja minimalni skup elemenata koje bi trebalo upotrebiti za sečenje, h_s je visina upotrebljenog elementa s , W i H su standardna širina, odnosno visina, w_i i h_i su širina, odnosno visina i -tog traženog elementa, W^* je širina kontejnera u koji su upakovani elementi, H^* je visina dela kontejnera minimalne povišine, koji obuhvata upakovane elemente, a P je koeficijent za penalizovanje prekoračenja širine standardnih elemenata, koji je u ovom radu postavljen na 100.

3.4. Selekcija

Implementirana su dva algoritma za selekciju jedinki, a to su ruletska selekcija sa linearnom normalizacijom i turnirska selekcija. U okviru ruletske selekcije sa linearnom normalizacijom, jedinke se sortiraju prema prilagođenosti, od najlošije do najbolje prilagođene. Zatim se simulira okretanje ruletskog točka onoliko puta koliko je jedinki potrebno selektovati. Pri tome je širina svakog polja na točku srazmerna rangui odgovarajuće jedinke. Turnirska selekcija podrazumeva održavanje turnira između q nasumično odabranih jedinki, onoliko puta koliko je potrebno odabrati jedinki za ukrštanje. U sklopu turnira, jedinke se porede po vrednosti funkcije troška. U ovom radu, veličina turnira je postavljena na 3.

Takođe, implementiran je elitistički model genetskog algoritma, koji nalaže da se iz svake generacije 2-5% najbolje prilagođenih jedinki direktno prenese u iduću generaciju, a da se ostale jedinke iz populacije zamene potomačkim jedinkama. Na ovaj način je osigurano to da genetski materijal najbolje prilagođenih jedinki neće biti izgubljen kroz proces smene generacija.

3.5. Ukrštanje

Od algoritama za ukrštanje, implementirani su parcijalno mapirano ukrštanje (eng. *Partially Mapped Crossover*), redosledno ukrštanje (eng. *Order Crossover*) i modifikovano redosledno ukrštanje. Modifikovano redosledno ukrštanje se razlikuje od redoslednog ukrštanja po tome što se geni iz segmenata roditeljskih jedinki ne kopiriraju na odgovarajuće pozicije potomačkih jedinki, već na njihov početak.

U slučaju da su jedinke reprezentovane kao izraz, prvo se geni razdvajaju prema tipu, odnosno operandi se grupišu odvojeno od operatora. Potom se uzimaju delovi sačinjeni od operanada roditeljskih jedinki, te se nad njima obavlja proces ukrštanja [3]. Nakon što se od dva roditeljska niza operanada dobiju dva potomačka, neophodno je spojiti ih sa operatorima roditeljskih jedinki, kako bi se dobile dve nove validne jedinke. Jedinke reprezentovane kao permutacija sadrže gene koji su svi istog tipa. Zbog toga, u ovom slučaju nije potrebno implementirati dodatne korake koji prethode samom algoritmu ukrštanja, ili koji mu slede.

3.6. Mutacija

U slučaju reprezentacije jedinke kao izraza, kao standardni mehanizam mutacije je implementirana dvotačkasta zamena gena u jedinki (eng. *Two point swap*), sa mogućnošću invertovanja pojedinačnih gena. Invertovanje gena se može odnositi na promenu operatora ili na promenu predznaka operanda, u zavisnosti od toga da li je posmatrani gen operator ili operand. Pri tome je neophodno osigurati to da mutirana jedinka predstavlja validan izraz. U slučaju reprezentacije jedinke kao permutacije, implementirana su dva standardna algoritma za mutaciju, a to su dvotačkasta zamena i trotačkasta zamena gena u jedinki (eng. *Three point swap*), uz invertovanje gena. Pri tome se pod invertovanjem gena smatra samo promena predznaka rednog broja kojim je označen posmatrani gen.

Takođe, implementirana je varijacija genetskog algoritma, u okviru koje su standardni mehanizmi mutacije zamenjeni tabu pretragom. Tabu pretraga (eng. *Tabu search*) je metaheuristička tehnika koja omogućava pretraživanje prostora rešenja. Počinje od jedne, početne jedinke, te identifikuje njoj susedne jedinke. U svakoj iteraciji tabu pretrage, bira se najbolje prilagođena jedinka iz susedstva koja nije dotad posećena, kako bi se sprečili ciklusi u pretrazi. Zatim se odabrana jedinka markira posećenom, tako što se dodaje u listu tabua, i postavlja se kao novo najbolje rešenje ukoliko ima manju vrednost funkcije troška od dotad najboljeg rešenja. Na kraju, najbolje prilagođena jedinka iz susedstva se bira kako bi zamenila posmatranu jedinku [5]. Kao susedstvo jedinke za potrebe tabu pretrage, posmatraju se jedinke koje su dobijene od posmatrane jedinke zamenom mesta dva nasumično odabrana gena ili invertovanjem gena. Zatim se od svih generisanih suseda nasumično bira njih 25 koji će biti posmatrani u sklopu tabu pretrage, kako bi se ograničilo vreme izvršavanja algoritma.

4. EVOLUCIONO PROGRAMIRANJE

Evoluciono programiranje (eng. *Evolutionary programming*) je još jedan optimizacioni algoritam iz porodice evolutivnih algoritama. Značajna razlika u odnosu na genetski algoritam je to što se u sklopu evolucionog programiranja ne primenjuje korak ukrštanja jedinki, već je mutacija jedini mehanizam pretrage prostora rešenja. Kao način za reprezentaciju jedinke, odabrane su iste dve strukture kao i u slučaju genetskog algoritma. Takođe, odabrani su isti mehanizmi za generisanje inicijalne populacije, evaluaciju i mutaciju. Od mehanizama mutacije implementiranih u sklopu genetskog algoritma, jedino tabu pretraga nije implementirana u sklopu evolucionog programiranja, kako algoritam ne bi zahtevao previše vremena za izvršavanje.

4.1. Selekcija jedinki za prelazak u narednu generaciju

Mehanizam koji se razlikuje u odnosu na genetski algoritam jeste selekcija. Implementirana su tri algoritma za selekciju jedinki koje će preći u narednu generaciju, a to su: turnirska selekcija sa razigravanjem, stohastičko univerzalno uzorkovanje i stohastičko univerzalno uzorkovanje sa linearnom normalizacijom.

U sklopu turnirske selekcije sa razigravanjem, svaka jedinka se takmiči sa 25 nasumično odabranih protivničkih jedinki. Potom se jedinke sortiraju po broju pobeda, i

odabira se onoliko jedinki sa najviše pobeda koliko je potrebno da pređe u narednu generaciju. Stohastičko univerzalno uzorkovanje predstavlja modifikaciju ruletske selekcije. Umesto simuliranja okretanja ruletskog točka sa jednim pokazivačem, onoliko puta koliko je potrebno selektovati jedinki, simulira se okretanje ruletskog točka koji ima onoliko pokazivača koliko je jedinki potrebno selektovati, jedanput. Pri tome je veličina segmenata na ruletskom točku obrnuta srazmerna vrednosti funkcije troška jedinki [6]. U okviru stohastičkog univerzalnog uzorkovanja sa linearnom normalizacijom, u obzir se uzima rang jedinke prema vrednosti funkcije troška, umesto same vrednosti funkcije troška. To znači da je veličina segmenata na ruletskom točku srazmerna rangu jedinki, a ne samoj vrednosti funkcije troška.

5. REZULTATI

U ovom poglavlju su analizirani i upoređeni rezultati primene genetskog algoritma i algoritma evolucionog programiranja za rešenje problema sečenja dvodimenzionalnih elemenata. Pri evaluaciji kvaliteta rešenja, posmatrani su sledeći kriterijumi:

- Prosečan broj dostupnih elemenata koje je neophodno upotrebiti (posmatra se razlika u odnosu na teoretski minimum)
- Prosečan procenat dobijenog otpada
- Prosečan broj iteracija algoritma
- Prosečno vreme izvršavanja algoritma

Implementirani algoritmi su podeljeni u četiri grupe:

1. Genetski algoritam, jedinka reprezentovana kao izraz
2. Genetski algoritam, jedinka reprezentovana kao permutacija
3. Evoluciono programiranje, jedinka reprezentovana kao izraz
4. Evoluciono programiranje, jedinka reprezentovana kao permutacija

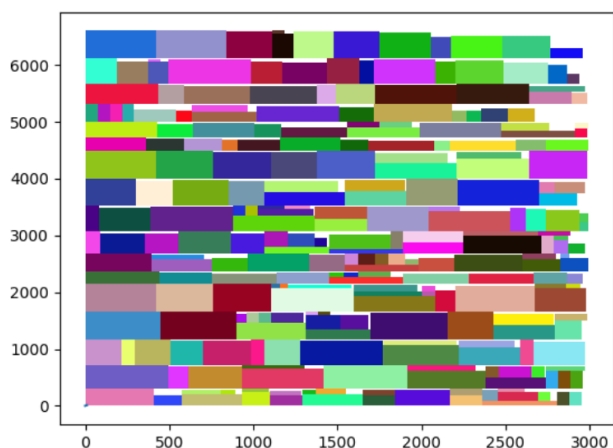
Zatim su svi algoritmi evaluirani nad skupom od 30 problema za upoređivanje iz [7], kako bi se identifikovao najbolji algoritam u svakoj grupi, te i najbolji algoritam sveukupno. Dobijeni rezultati su prikazani u tabeli 1.

Tabela 1. Prosečni rezultati dobijeni primenom najboljih algoritama iz grupa 1-4

	Grupa 1	Grupa 2	Grupa 4	Grupa 4
Broj elemenata za sečenje	1.43	0.43	2.02	0.47
Procenat optada	0.22	0.18	0.23	0.20
Broj iteracija	204.77	141.6	1189.15	493.73
Vreme izvršavanja (s)	11847.9	4844.7	232.1	952.6

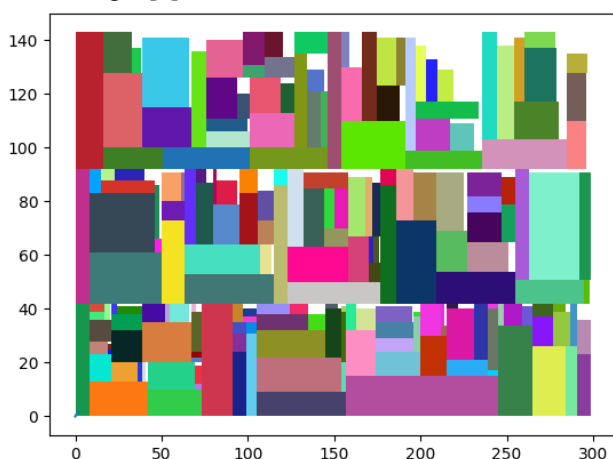
Nakon što su relevantni kriterijumi uzeti u obzir, kao najbolji je odabran algoritam evolucionog programiranja, u okviru kog su jedinke reprezentovane kao permutacija, koji kao mehanizam selekcije implementira turnirsku selekciju, a kao mehanizam mutacije implementira višestruku dvotačkastu zamenu gena, uz mogućnost invertovanja gena.

Algoritam je dalje evaluiran nad skupom problema iz [8], koji nalažu sečenje između 250 i 2700 elemenata. Dobijeni rezultati su pokazali da se odstupanje od teoretskog optimuma za broj upotrebljenih elemenata za sečenje kreće u opsegu između 0 i 5, a da se procenat dobijenog otpada kreće se u opsegu između 6.8 i 10. Na slici 1 je prikazan primer rešenja problema AB_5 iz skupa problema za upoređivanje [8].



Slika 1. Primer rešenja problema AB_5 iz skupa [8]

Takođe, algoritam je primenjen nad skupom problema koji zahtevaju sečenje između 17 i 199 elemenata, nad kojima je evaluiran i algoritam implementiran u radu [2]. Pokazalo se da odstupanje od rezultata iz rada [2], u prosečnim dobijenim vrednostima, nije veće od 3%. Na slici 2 je prikazan primer rešenja problema T7c, iz skupa problema navedenog u [2].



Slika 2. Primer rešenja problema T7c iz skupa problema datom u [2]

6. ZAKLJUČAK

U ovom radu je predstavljeno rešenje problema sečenja dvodimenzionalnih elemenata, sa rezovima od ivice do ivice, pomoću genetskog algoritma i algoritma

evolucionog programiranja. Nakon evaluacije algoritama nad skupom problema za upoređivanje, kao najbolji je odabran algoritam evolucionog programiranja, u okviru kog su jedinke reprezentovane kao permutacija, koji kao mehanizam selekcije implementira turnirsku selekciju, a kao mehanizam mutacije implementira višestruku dvotačkastu zamenu gena, uz mogućnost invertovanja gena. Algoritam se dobro pokazao i prilikom rešavanja problema sečenja do 2700 elemenata, pri čemu maksimalno odstupanje od teoretskog optimuma iznosi 5. Takođe, algoritam je upoređen sa rezultatima iz rada [2], te je u proseku dobijeno odstupanje ne veće od 3% od rezultata iz tog rada.

Rešenje problema sečenja dvodimenzionalnih elemenata bi se moglo generalizovati dozvoljavanjem upotrebe dostupnih elemenata različitih dimenzija. Takođe, bilo bi korisno uzeti u obzir redosled sečenja elemenata, kako bi se minimizovao broj zamena na mašini za sečenje, a samim tim i vreme utrošeno na sečenje.

7. LITERATURA

- [1] Mellouli, A. and Dammak, A., 2008. An algorithm for the two-dimensional cutting-stock problem based on a pattern generation procedure. *Inf Manag Sci*, 19(2), pp.201-218.
- [2] Gharsellaoui, H. and Hasni, H., 2011. An hybrid genetic algorithm for two-dimensional cutting problems using guillotine cuts. In *Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- [3] Ono, T. and Ikeda, T., 1999, July. Optimizing two-dimensional guillotine cut by genetic algorithms. In *Proc. of the Ninth AJOU-FIT-NUST Joint Seminar* (pp. 40-47).
- [4] Jylänki, J., 2010. A thousand ways to pack the bin-a practical approach to two-dimensional rectangle bin packing. retrieved from <http://clb.demon.fi/files/RectangleBinPack.Pdf>
- [5] Glover, F., 1990. Tabu search: A tutorial. *Interfaces*, 20(4), pp.74-94.
- [6] Jebari, K. and Madiafi, M., 2013. Selection methods for genetic algorithms. *International Journal of Emerging Sciences*, 3(4), pp.333-344
- [7] <https://github.com/Oscar-Oliveira/OR-Datasets/tree/master/Cutting-and-Packing/2D/Datasets/C>
- [8] <https://github.com/Oscar-Oliveira/OR-Datasets/tree/master/Cutting-and-Packing/2D/Datasets/AB>

Kratka biografija:



Milica Popović je rođena 23.11.2000. godine u Banjaluci. Osnovne akademske studije na Fakultetu tehničkih nauka, smer Softversko inženjerstvo i informacione tehnologije, završila je 2023. godine. Master akademske studije na istom smeru je završila 2024. godine. kontakt: milica.popovic720@gmail.com