

12-30-2011

Optimizacija procesa razreza po skupinah

Mihael Cesar

Mirko Gradišar

Jure Erjavec

Luka Tomat

Follow this and additional works at: <https://www.ebrjournal.net/home>

Recommended Citation

Cesar, M., Gradišar, M., Erjavec, J., & Tomat, L. (2011). Optimizacija procesa razreza po skupinah. *Economic and Business Review*, 13(5). <https://doi.org/10.15458/2335-4216.1229>

This Original Article is brought to you for free and open access by Economic and Business Review. It has been accepted for inclusion in Economic and Business Review by an authorized editor of Economic and Business Review.

OPTIMIZACIJA PROCESA RAZREZA PO SKUPINAH

MIHAEL CESAR¹
MIRKO GRADIŠAR²
JURE ERJAVEC³
LUKA TOMAT⁴

POVZETEK: V članku je predlagana metoda reševanja enodimenzionalnega problema razreza materiala po skupinah. Metoda je uporabna v primeru velikih naročil, ki jih zaradi tehnoloških in logističnih razlogov ni mogoče izpolniti v enem, temveč v več zaporednih korakih. Metoda ima dve stopnji. V prvi stopnji se izračunajo pari naročil, ki se v drugi stopnji združujejo v skupine. Metoda se izvaja v obliki računalniškega algoritma G-CUT in je preskušena na primeru iz prakse.

Ključne besede: operacijske raziskave, optimizacija, razrez

JEL klasifikacija: C61

1. UVOD

Enodimenzionalni razrez materiala nastopa v mnogih panogah od lesne (Venkateswarlu, 2001; Manrique et al., 2011), papirne (Chauhan, Martel, & D'amour, 2008; Matsumoto, Umetani & Nagamochi, 2011), tekstilne (Gradišar, Jesenko, & Resinovič, 1997) do kovinske (Cui, Gu, & Hu, 2009). Vsem tem panogam je skupen problem, kako iz daljših v splošnem različnih palic na zalogi narezati naročeno število krajših palic. Pri optimizaciji enodimenzionalnega razreza želimo najti odgovor na vprašanje, kako izvesti razrez, da bo čim manj izgub materiala oziroma ostankov, ki jih zaradi premajhnih in neuporabnih dimenzij zavržemo (Kantorovich, 1960; Gilmore & Gomory, 1961, 1963, 1965; Gass, 1985; Gradišar, 1999; Erjavec, Gradišar, Trkman, 2010; Gradišar, Erjavec, Tomat, 2011;).

Zmanjšanje neuporabnega ostanka oz. izgube materiala je večinoma glavni cilj optimizacije, vendar ni edini, kajti zaradi kombinacije tehnoloških, proizvodnih in poslovnih

¹ Univerza v Ljubljani, Ekonomska fakulteta, Ljubljana, e-naslov: mihael.cesar@ef.uni-lj.si

² Univerza v Ljubljani, Ekonomska fakulteta, Ljubljana, e-naslov: miro.gradisar@ef.uni-lj.si

³ Univerza v Ljubljani, Ekonomska fakulteta, Ljubljana, e-naslov: jure.erjavec@ef.uni-lj.si

⁴ Univerza v Ljubljani, Ekonomska fakulteta, Ljubljana, e-naslov: luka.tomat@ef.uni-lj.si

značilnosti, ki so od primera do primera različne, je treba upoštevati tudi dodatne omejitve. Vsaka dodatna omejitev sicer zmanjša število možnih rešitev in s tem teoretično skrajša čas iskanja najboljše, kljub temu pa je čas za iskanje najboljše rešitve, ki je matematično opredeljena kot globalni minimum kriterijske funkcije, pri obsežnejših primerih predolg in pomeni ozko grlo za poslovni proces v podjetju. Razlog za časovno potratnost pri iskanju najboljše rešitve tiči v dejstvu, da je večina problemov razreza NP-polnih (Bischoff & Wäscher, 1995). Uporaba metod, ki najdejo zadovoljiv rezultat v določenem časovnem intervalu, je zato ključna za odpravljanje ozkih grl, čeprav vedno ne najdejo najboljše rešitve, je pa rešitev blizu optimalni.

V praksi se podjetja srečujejo s specifičnimi problemi, kjer standardne metode ne pridejo v poštev, temveč jih je treba prilagoditi ali pa razviti nove. Kljub dokaj jasni definiciji problema enodimenzionalnega razreza v poslovnem svetu obstaja veliko število raznovrstnih primerov. Poleg zmanjšanja izgube materiala so cilji lahko tudi nižji stroški, krajši čas razreza, nižje število menjav ali nastavitev rezila, zmanjšanje zaloge, prilagoditev razreza za potrebe logistike in podobno (Trkman & Gradišar, 2003).

Poseben izziv so obsežnejši primeri naročil, ki jih sestavlja veliko število kosov. V literaturi meja, ki loči manj in bolj obsežna naročila, ni jasno določena. Poleg števila kosov določajo velikost naročila tudi dimenzije, oblika profila, teža materiala, logistične in tehnološke omejitve, ki izhajajo iz specifičnih pogojev v praksi. Osnovni problem je ta, da materiala iz skladišča za celotno naročilo ni možno hkrati prepeljati do stroja za rezanje. Tudi prostor za odlaganje materiala ob stroju na eni strani in narezanih kosov na drugi je navadno omejen. Poleg tega je treba narezane kose odlagati tako, da jih je kasnejše čim lažje jemati s kupa oziroma sklada v vrstnem redu, ki ga zahtevajo nadaljnji tehnološki postopki in ki je praviloma različen od vrstnega reda rezanja, ki ga določa algoritem za minimizacijo ostanka.

V literaturi še ni zaslediti niti splošnih metod za optimizacijo razreza pri velikih naročilih niti rešitve konkretnih praktičnih primerov. V tem prispevku sicer opisujemo razvoj metode, ki je prilagojena potrebam konkretnega podjetja, vendar pa je sorazmerno malo specifičnih omejitev. Zato lahko sklepamo, da bo predlagana metoda lahko služila kot vzorec rešitve tudi drugim podjetjem, ki se srečujejo s problemom velikih naročil.

Predlagana metoda predvideva razdelitev velikega naročila v več manjših delov, podnaročil oziroma skupin naročenih dolžin tako, da bi bila skupna izguba materiala vseh skupin najmanjša.

Nadaljevanje članka sestavlja pet razdelkov. V drugem poglavju je definicija problema. Tretje poglavje opisuje metodo, ki rešuje problem. V četrtem poglavju sledi testiranje metode z realnimi podatki. Zadnje poglavje pa vsebuje sklepne misli in predloge nadaljnega raziskovanja na tem področju.

2. DEFINICIJA PROBLEMA

Na zalogi imamo dovolj kosov enodimenzionalnega materiala – palic, da izpolnimo naročilo. Dolžina posamezne palice je v splošnem različna in izražena s celimi števili. Palice so lahko enake ali različne. Različne so zaradi tega, ker proizvajalec izdeluje več različnih standardnih dolžin, ker so pomembne razlike med deklariranimi standardnimi dolžinami in dejansko izmerjenimi in ker obstajajo tudi nestandardne dolžine, ki so lahko dovolj veliki oziroma še uporabni ostanki prejšnjih naročil.

Material iz zaloge uporabimo za izpolnitev naročila, ki ga sestavljajo krajše dolžine s točno določenim številom kosov. Zaradi velikosti naročilo razdelimo na r manjših skupin naročenih dolžin. r ni konstanta, ker morata biti tako velikost kot sestava posamezne skupine v skladu z omejitvami. Gre za obraten problem kot pri združevanju naročil, ki morajo biti na primer izpolnjena do določenega datuma (Li, 1996), in podoben kot pri izpolnjevanju naročil v zaporednih časovnih intervalih (Trkman & Gradišar, 2007), le da se v našem primeru zaloga materiala med izpolnjevanjem naročila ne spreminja. Ostanek, ki je dovolj dolg, da ga bomo lahko uporabili pri izpolnjevanju prihodnjih naročil, vrnemo v skladišče (Alferi, 2004; Cherri, Areales, Yanasse, 2009; Gradišar, Kljajić, Resinovič, Jesenko, 1999). Problemi razreza, kjer je predvidena ponovna uporaba dovolj dolgih ostankov, se označujejo z angleško kratico CSPUL (Cutting Stock Problems with Usable Leftovers), ostanki, krajši od določenega praga D , pa predstavljajo izgubo. Naročilo izpolnimo tako, da minimiziramo skupno izgubo vseh skupin.

Kriterijsko funkcijo in omejitve bomo sicer oblikovali v skladu s potrebami konkretnega podjetja, ki se ukvarja z izdelavo stebrov za visokonapetostne električne vode. Vendar pa je večina omejitev precej splošnih in je opisan problem najbrž podoben, kot ga imajo tudi druga podjetja, ki se srečujejo z velikimi naročili.

Treba je minimizirati izgubo materiala celotnega naročila tako, da

1. se posamezna naročena dolžina nahaja samo v eni skupini, ker je rokovanje z narezanimi kosi ob stroju za rezanje tako mnogo lažje;
2. se lahko posamezna palica v skladišču uporabi samo v eni skupini, ker je rokovanje z ne do konca razrezanimi palicami tako mnogo lažje;
3. je v posamezni skupini največ P naročenih dolžin, kar olajšuje rokovanje z narezanimi kosi ob stroju za rezanje in tudi izbiranje palic, ki gredo v nadaljnji tehnološki postopek;
4. se lahko iz posamezne palice v skladišču reže največ N različnih naročenih dolžin, ker s tem omejimo število nastavitvev dolžine rezanja na stroju in olajšamo rokovanje z delno razrezanimi palicami iz skladišča;
5. je velikost skupine omejena z M – največjo dovoljeno vsoto dolžin naročenih kosov v skupini. To olajšuje prevoz palic iz skladišča do stroja za rezanje, tako da se ena skupina prepelje hkrati in olajšuje rokovanje s palicami ob stroju za rezanje.

Problem razreza takole definiramo:

l_i = dolžina naročene palice; $i = 1, \dots, n$

b_i = zahtevana količina palic dolžine l_i

L_j = dolžina palice na zalogi; $j = 1, \dots, m$

x_{ij} = število naročenih kosov dolžine l_i , ki so odrezani iz L_j

∂_j = ostanek pri razrezu palice L_j

Kriterijska funkcija:

$$\min \sum_{k=1}^r \sum_{j=1}^m t_{kj} \cdot g_{kj} \quad (\text{minimiziranje izgube celotnega naročila})$$

z ozirom na:

- 1) $\sum_{i=1}^n l_i h_{kj} \leq M \quad \forall k$ (največja vsota dolžin vseh naročil v skupini)
- 2) $\sum_{k=1}^r g_{kj} \leq 1 \quad \forall j$ (vsaka palica v skladišču se lahko uporabi le v eni skupini)
- 3) $\sum_{k=1}^r h_{kj} \leq 1 \quad \forall i$ (vsako naročilo je lahko samo v eni skupini)
- 4) $\sum_{i=1}^n h_{kj} \leq P \quad \forall k$ (največje število različnih naročil v skupini)
- 5) $\sum_{i=1}^n y_{ij} \leq N \quad \forall j$ (z ene palice na skladišču lahko odrežemo le N različnih naročil)
- 6) $\sum_{i=1}^n l_i c_{ij} + \partial_j = L_j \quad \forall j$ (omejitev nahrbtnika)
- 7) $\sum_{j=1}^m x_{ij} = b_i \quad \forall i$ (naročene dolžine so proizvedene v točno zahtevanem številu)
- 8) $\sum_{j=1}^m z_j \cdot u_{kj} \leq 1 \quad \forall k$ (samo en ostanek v skupini k je lahko daljši od $\max(l_i, h_{ki})$)
- 9) $UB_k = \max(l_i \cdot h_{ki}) \quad \forall k$
 $x_{ij} \geq 0$, celo število $\quad \forall i, j$
 $t_{kj} \geq 0 \quad \forall j$
 $\partial_j \geq 0 \quad \forall j$.

V navedenem modelu so uporabljene naslednje funkcije:

$$z_j = \begin{cases} 0 & \text{v primeru, ko } x_{ij} = 0 \quad \forall i \\ 1 & \text{sicer} \end{cases}$$

kaže, če je palica L_j uporabljena v načrtu razreza,

$$g_{kj} = \begin{cases} 1 & \text{v primeru, ko je } L_j \text{ uporabljen v skupini } k \\ 0 & \text{sicer} \end{cases}$$

kaže, če je palica L_j uporabljena v skupini k ,

$$h_{ki} = \begin{cases} 1 & \text{v primeru, ko je } l_i \text{ v skupini } k \\ 0 & \text{sicer} \end{cases}$$

kaže, če je naročilo l_i v skupini k ,

$$u_{kj} = \begin{cases} 1 & \text{v primeru, ko } g_{kj} = 1 \wedge \partial_j > UB_k \\ 0 & \text{sicer} \end{cases}$$

kaže, če je ostanek palice L_j , uporabljene v skupini k , večji od UB_k ,

$$t_{kj} = \begin{cases} \partial_j & \text{v primeru, ko } g_{kj} = 1 \wedge \partial_j \leq UB_k \\ 0 & \text{sicer} \end{cases}$$

predstavlja ostanek palice L_j v skupini k ,

$$y_{ij} = \begin{cases} 0 & \text{v primeru, ko } x_{ij} = 0 \\ 1 & \text{sicer} \end{cases}$$

kaže, če se naročilo l_i reže iz palice L_j .

V vsaki skupini je lahko le en ostanek daljši od $\max(l_i, h_{ki})$ (8). Ostanki v skupini k , ki so daljši od $\max(l_i, h_{ki})$, se ne štejejo kot izguba. Če bi bila zgornja meja dolžine ostankov v posamezni skupini UB_k , ki jih smatramo kot izgubo (ker so prekratki, da bi jih lahko ponovno uporabili), postavljena nižje od $\max(l_i, h_{ki})$, bi prišlo do naslednje situacije. Ob dovolj veliki zalogi bi algoritem generiral čim več ostankov, ki so manjši od $\max(l_i, h_{ki})$ in večji od UB_k , ker se ne bi šteli kot izguba. Na zalogi bi se tako nabiralo vedno večje število krajših kosov, ki bi povzročali večji ostanek pri prihodnjih naročilih. To bi se zgodilo tudi, če bi dovolili več kot en ostanek na skupino, daljši od $\max(l_i, h_{ki})$. Ker tega ne želimo, nastavimo UB_k na $\max(l_i, h_{ki})$, po izvedeni optimizaciji pa lahko poljubno določimo mejo D , ki ločuje izgubo od še uporabnih ostankov, ki jih vrnemo v skladišče.

3. RAZVOJ REŠITVE

Skupine lahko v splošnem tvorimo na dva načina. Lahko optimiziramo naročilo kot celoto, rezultate pa potem razdelimo v manjše skupine. V tem primeru posebne metode za optimizacijo razreza ne potrebujemo, ker skupine oblikujemo, ko je načrt razreza že izdelan. Lahko pa najprej oblikujemo skupine in nato optimiziramo razrez. Če bi uporabili prvi način, v splošnem ne bi mogli zadovoljiti omejitve (3). Ker iz posamezne palice v skladišču

lahko odrežemo kose katerih koli naročenih dolžin, grupiranje palic, ki vsebujejo samo določena naročila, ne bi bilo možno. Razviti je torej treba novo metodo, ki bo omogočala obraten postopek, najprej oblikovanje skupin in nato minimizacijo ostanka. Ključna za obstoj problema optimizacije razreza po skupinah je torej omejitev (3). Od splošne definicije CSPUL (Gradišar et al., 1999) se naša definicija razlikuje le po tem, da ima poleg omejitve velikosti skupine (1) ali (4) samo dve dodatni omejitvi: (3) in (2). Omejitev (2) ni tako pomembna, lahko bi jo odpravili, a to ne bi pomembno vplivalo na predlagano metodo reševanja problema. Gre za to, da uporabni ostanki prejšnjih skupin niso na voljo naslednjim skupinam istega naročila, ampak se po izpolnitvi naročila vrnejo v skladišče. V dani praktični situaciji se je takšno obravnavanje uporabnih ostankov izkazalo kot primerno, v kakšnem drugem podjetju pa omejitev (2) morda ne bi bila potrebna.

Predlagana hevristična metoda ima dva dela. V prvem delu celotno naročilo razdelimo v skupine, v drugem delu pa minimiziramo ostanke posameznih skupin.

Delitev večjega naročila na manjše dele je možno izvesti na mnogo načinov. Izbrali bomo tistega, ki je primernejši za reševanje naročil, ki jih izpolnjujemo z velikim številom palic iz skladišča. To se zgodi v primerih z majhnim razmerjem med povprečno dolžino palice v skladišču in povprečno dolžino naročila. To razmerje je majhno, če je na primer med 1 in 3. V teh primerih je število palic iz skladišča, ki jih potrebujemo za izpolnitev naročila, zelo veliko oziroma le malo manjše, kot je število kosov naročil. V nasprotnem primeru, ko je to razmerje večje, je možno naročilo izpolniti že z manjšim številom palic iz skladišča. S tem pa se v splošnem tudi zmanjša potreba po delitvi naročila na lažje obvladljive dele. Pri oblikovanju skupin torej domnevamo, da je večina praktičnih potreb po razdelitvi naročila na manjše dele povezana s problemi, pri katerih iz ene palice v skladišču odrežemo kose le enega ali najpogosteje dveh različnih naročil, več pa le izjemoma. Takšno stanje je tudi v obravnavanem podjetju, zato smo razvoj nove metode prilagodili majhnim razmerjem med povprečno dolžino palic v skladišču in dolžino naročil. V praktičnih primerih, kjer je to razmerje večje od 3, zato predlagane metode ne bo možno uporabiti brez večjih prilagoditev. Predvidevamo pa lahko, da takšnih primerov v praksi ni veliko.

Naročilo razdelimo v skupine v dveh stopnjah. V prvi iz naročil tvorimo pare, ki predstavljajo najmanjše možne skupine, v drugi pa iz parov oblikujemo končne skupine.

Metoda temelji na predpostavki, da je najbolje, če v postopku razreza damo prednost daljšim dolžinam. Tako zmanjšamo učinek ostrejših pogojev proti koncu postopka (ending conditions) (Gradišar, 1999; Trkman & Gradišar, 2007) in povečamo verjetnost, da bomo našli zadovoljivo rešitev. V ta namen najprej tvorimo seznam naročil S_n , urejen po padajočih dolžinah, nato izberemo prvo oziroma najdaljšo dolžino iz seznama S_n in ji določimo par. Če ima seznam S_n liho število naročil, ga dopolnimo do sodega števila tako, da mu na koncu dodamo še eno naročilo z dolžino *maxint* in številom kosov 1. *maxint* je največje celo število, ki je lahko predstavljeno v računalniku. Pri enojni natančnosti je to 32.767. Ker tako velikega naročila ne moremo realizirati oziroma kombinirati z drugimi naročili, na razrez ne vpliva. S tem smo zagotovili, da bo imelo vsako naročilo svoj par.

Par določimo tako, da za vse možne kombinacije najdaljše dolžine s seznama S_n z ostalimi krajšimi dolžinami rešimo problem nahrbtnika ob upoštevanju vse razpoložljive zaloge. Izberemo tisti par, ki ima najmanjši povprečni ostanek. Povprečni ostanek izračunamo kot kvocient vsote ostankov uporabljenih palic iz skladišča, zmanjšane za ostanke, ki jih vrnemo v skladišče, in števila uporabljenih palic iz skladišča, zmanjšane za število ostankov, ki smo jih vrnili v skladišče. Ko je par izbran, s seznama S_n črtamo obe dolžini, ki ga sestavljata, iz zaloge pa tiste palice, ki smo jih za ta par porabili skupaj z vrnjenimi ostanki. Postopek določanja parov ponavljamo, dokler so v seznamu S_n še razpoložljive dolžine. Rezultat tega postopka je $n/2$ parov. Vsakemu paru je pridružen povprečni ostanek.

Namesto parov bi lahko tvorili tudi podskupine iz večjega števila naročil, tako da bi bilo to čim bolj usklajeno s številom N pri omejitvi (5). Tudi predlagana metoda se v osnovi ne bi spremenila. Za pare smo se odločili, ker v primerih z majhnim razmerjem med povprečno dolžino palice v skladišču in povprečno naročeno dolžino kaj drugega ne bi imelo smisla in ker je število možnih kombinacij, ki jih moramo preveriti, najmanjše. Poleg tega imamo v nadaljevanju več možnosti pri oblikovanju končnih skupin.

Dobljene pare uredimo po padajočem povprečnem ostanku v seznam S_r , S_r nato razdelimo na skupine parov tako, da se pomikamo od prvega proti zadnjemu. V vsaki skupini, razen morda v zadnji, je ravno toliko parov, da ne presežemo omejitve P ali M . Dobljene skupine parov v nadaljevanju obravnavamo kot samostojna naročila. Povezuje jih skupna zaloga palic, zato ni vseeno, v kakšnem vrstnem redu se bo izvajal razrez posameznih skupin. Prva skupina bo imela na voljo celotno zalogo palic. Tako bo pri optimizaciji razreza na voljo več možnih kombinacij, zato lahko pričakujemo manjši ostanek (Gradišar, 1999). Obratno pa bo pri optimizaciji kasnejših skupin na voljo samo še toliko zaloge, kot je ostalo od razreza prejšnjih skupin. Možnih kombinacij bo torej manj, pričakovan ostanek pa večji.

V drugem delu najprej določimo vrstni red razreza posameznih skupin. Pri tem izhajamo iz predpostavke, da je bolje, če pri razrezu damo prednost skupinam s pari, ki so višje na seznamu S_r in imajo torej višji pričakovani ostanek. Domnevamo, da bi se ostanek bolj povečal, če bi te skupine razrezali na koncu, kot bi se povečal, če na koncu režemo skupine z nižjim pričakovanim ostankom. Vrstni red razreza skupin je torej enak kot vrstni red njihovega tvorjenja. Optimizacijo razreza posamezne skupine lahko izvedemo po kateri koli metodi za reševanje CSPUL. Izbrali smo metodo C-CUT (Gradišar & Trkman, 2005) v obliki programske rešitve, ki omogoča nastavljanje največjega števila naročenih dolžin, ki se režejo iz posamezne palice na zalogi in ki uporablja eksaktno metodo za manj obsežne skupine.

Rezultat drugega dela je načrt razreza, ki vsebuje dve vrsti ostankov. Manjši od D so izguba, daljši pa se vrnejo v skladišče. Na sliki 1 sta oba dela predlagane metode predstavljeni v obliki psevdokode.

Prvi del: določitev parov

Razvrsti naročila po padajočih dolžinah v $S_n (l_1 > l_2 > \dots > l_i > \dots > l_n)$.

$j \leftarrow 1, \dots, n-1; k \leftarrow 2, \dots, n$

$R_{j,k} \leftarrow \maxint \forall j, k$ (začetne vrednosti povprečnih ostankov vseh parov)

$i \leftarrow 1$ (začetna vrednost števca i)

$h \leftarrow 1$ (začetna vrednost indeksa parov z minimalnimi ostanki)

ponavljaj dokler $i < n-1$

$p \leftarrow i+1$ (začetna vrednost števca p)

$R_{min} \leftarrow \maxint$ (začetna vrednost minimalnega ostanka)

$c \leftarrow 0$ (začetna vrednost testne spremenljivke c)

ponavljaj dokler $p < n$

če $l_i > 0 \wedge l_p > 0$ **potem**

Izvedi proceduro KNAPSACK⁵ za l_i in l_p .

Izračunaj povprečni ostanek R .

če je $R < R_{min}$, **potem**

$R_{min} \leftarrow R$

$j \leftarrow i$

$k \leftarrow p$

$c \leftarrow 1$ (c testira, ali je vsaj enkrat veljalo $R < R_{min}$)

konec če

konec če

$p \leftarrow p+1$

konec ponavljanj

če $c = 1$ **potem**

$R_h \leftarrow R_{min}$

$s1_h \leftarrow j$

$s2_h \leftarrow k$

$l_j \leftarrow 0; l_k \leftarrow 0$ (iz zaloge odstranimo palice, ki jih je porabil par l_j in l_k)

$h \leftarrow h+1$

konec če

$i \leftarrow i+1$

konec ponavljanj

Drugi del: oblikovanje skupin

Razvrsti povprečne ostanke R_h in pripadajoče številke parov po padajoči vrednosti R_h v sezname $S_r (R_1 > R_2 > \dots > R_{n/2})$, $S_{p1} (s1_1, s1_2 \dots s1_{n/2})$ in $S_{p2} (s2_1, s2_2 \dots s2_{n/2})$.

$h \leftarrow 1$ (začetna vrednost števca parov)

$a \leftarrow 1$ (trenutno število različnih naročil v skupini)

$b \leftarrow 0$ (trenutna skupna dolžina naročil v skupini)

$gl \leftarrow 1$ (trenutna spodnja meja skupine)

⁵ Procedura KNAPSACK je enaka kot v (Gradišar et al., 1999).

ponavljalj dokler $h \leq n/2$

$i \leftarrow g1$

ponavljalj dokler $i \leq h$

$a \leftarrow a + 2$

$b \leftarrow l_{s1i} \cdot b_{s1i} + l_{s2i} \cdot b_{s2i}$

$i \leftarrow i + 1$

konec ponavljanj

če $a > P \vee b > M$ **potem**

$g2 \leftarrow i - 2$ (trenutna zgornja meja skupine)

če $g1 < g2$, **potem**

$g1 \leftarrow g2$ (Če že en sam par preseže omejitve P ali M , ga vseeno razrežemo.)

konec če

Izvedi optimizacijo razreza skupine parov $s1_{g1}$ do $s1_{g2}$ in $s2_{g1}$ do $s2_{g2}$ s programom C-CUT⁶.

$g1 \leftarrow g2 + 1$

$h \leftarrow h - 1$

konec če

$h \leftarrow h + 1$

konec ponavljanj

Slika 1: *Algoritem razreza po skupinah, izražen v psevdokodi*

Optimizacija razreza skupin je potrebna zaradi omejitve (8), po kateri je v vsaki skupini le en ostanek lahko daljši od $\max(l_i, h_{ki})$. Brez te omejitve bi lahko uporabili že optimizirane pare, ki bi jih združevali v skupine.

Predlagana metoda je implementirana v obliki računalniškega programa G-CUT, ki omogoča reševanje problema, definirane v drugem razdelku, po predlagani metodi. Numeričen del je napisan v programskem jeziku Fortran, ki omogoča hitro procesiranje, vhod in izhod pa sta izvedena v 4GL (angl. Fourth-generation programming language, označuje programski jezik četrte generacije). Kot je razvidno iz slike 1, program G-CUT vsebuje tudi program C-CUT, ki omogoča optimizacijo razreza posamezne skupine s kombinacijo eksaktnega in hevrističnega algoritma. Za eksaktno reševanje uporablja programsko rešitev CPLEX. Z eksaktnimi metodami lahko optimiziramo naročila, ki ne presegajo določenih omejitev (Belov & Sheithauer, 2002; Alves & Carvalho, 2008). Večje skupine pa C-CUT optimira s hevrističnim algoritmom (Gradišar & Trkman, 2005).

Omejitve programa G-CUT so:

število naročil < 500

število kosov posameznega naročila < 500

število dolžin na zalogi < 500

število kosov posamezne dolžine na zalogi < 1000

⁶ Program C-CUT je enak kot v (Gradišar & Trkman, 2005).

4. REZULTATI

Za ilustracijo delovanja programa G-CUT bomo predstavili optimizacijo konkretnega praktičnega primera enodimenzionalnega razreza jeklenih profilov. Primer ima naslednje omejitve: $P = 8$, $N = 2$, $M = 150.000$ in $D = 2.500$. Vse dimenzije so v milimetrih.

Vhodni podatki so v tabeli 1.

Tabela 1: *Palice na zalogi in naročila*

ZALOGA		NAROČILO	
Dolžina palice (mm)	Število kosov	Dolžina palice (mm)	Število kosov
12.965	7	9.940	4
11.965	10	9.450	2
10.965	37	8.480	2
6.945	2	7.530	2
6.465	4	6.910	4
		6.145	2
		6.000	4
		5.710	2
		5.600	12
		5.280	8
		4.825	12
		420	18

Dolžine naročila so v skladu s predlagano metodo urejene po velikosti od najdaljše do najkrajše.

Iz primera je razvidno, da naročilo brez vračunanih ostankov po skupni dolžini obsega 52 % materiala, ki je na zalogi. Razmerje med povprečno dolžino palice na strani zaloge in tisto na strani naročila je nizko in znaša 1,55. V povprečju se torej iz palic na skladišču odreže manj kot dva kosa naročila.

Rezultat prvega dela metode je seznam parov, urejenih po padajočem povprečnem ostanku. Ker je naročenih dolžin 12, je parov 6. Prikazani so v tabeli 2.

Tabela 2: *Razdelitev parov na skupine*

Št. para	1. naročilo (mm)	2. naročilo (mm)	povprečni ostanek (mm)	
1	7.530	4.825	1.113	prva skupina
2	8.480	5.280	1.098	
3	6.000	5.600	565	
4	9.450	6.910	528	
5	9.940	420	135	druga skupina
6	6.145	5.710	101	

Zaradi omejitve $P = 8$ so pari razdeljeni na dve skupini. Prvo skupino tvorijo prvi štirje pari z osmimi naročenimi dolžinami in drugo zadnja dva s štirimi. S programom C-CUT optimiziramo najprej prvo skupino, ki jo tvorijo pari z večjim povprečnim ostankom, vendar pa imamo pri tem na voljo celotno zalogo palic. Rezultati so v tabeli 3. Z ostankom palic pa optimiziramo drugo skupino. Rezultati so v tabeli 4.

Tabela 3: Načrt razreza prve skupine

PODATKI			
ZALOGA		NAROČILO	
Dolžina palice (mm)	Število kosov	Dolžina palice (mm)	Število kosov
12.965	7	7.530	2
11.965	10	4.825	12
10.965	37	8.480	2
6.945	2	5.280	8
6.465	4	6.000	4
		5.600	12
		9.450	2
		6.910	4
NAČRT REZANJA			
Dolžina zaloge (mm)	Št. kosov zaloge	Vzorec rezanja naročil	Ostane (mm)
6.945	2	1 x 6.910	35
12.965	2	1 x 7.530, 1 x 4.825	610
12.965	2	1 x 6.000, 1 x 6.910	55
10.965	8	1 x 5.280, 1 x 5.600	85
10.965	2	1 x 4.825, 1 x 6.000	140
10.965	4	1 x 4.825, 1 x 5.600	540
10.965	2	2 x 4.825	1.315
10.965	2	1 x 9.450	1.515
10.965	2	1 x 8.480	2.485

Tabela 4: Načrt razreza druge skupine

PODATKI			
ZALOGA		NAROČILO	
Dolžina palice (mm)	Število kosov	Dolžina palice (mm)	Število kosov
12.965	3	9.940	4
11.965	10	420	18
10.965	17	6.145	2
6.465	4	5.710	2
NAČRT REZANJA			
Dolžina zaloge (mm)	Št. kosov zaloge	Vzorec rezanja naročil	Ostane (mm)
11.965	2	1 x 6.145, 1 x 5.710	110
12.965	2	1 x 9.940, 7 x 420	85
10.965	2	1 x 9.940, 2 x 420	185

Iz rezultatov obeh skupin vidimo, da noben ostanek ni daljši od $\max(l_i, h_{ki})$, ki znaša v prvi skupini 9.450 in v drugi 9.940. Vsi ostanki štejejo kot izguba, ker so manjši od 2.500.

Prikazani primer je sicer vzet iz realnosti, vendar pa predstavlja izjemno majhno naročilo. Izbrali smo ga zato, ker je zaradi preprostosti lažje razumljiv in s tem primernejši za predstavitev metode. Ker je naročilo majhno in sta tudi obe skupini sorazmerno majhni, sta bili obe s programom C-CUT rešeni eksaktno. Za rešitev prikazanega primera je osebni računalnik (Intel Core 2 Duo) potreboval 2,5 sekunde.

Podobno kot v (Gilmore & Gomory, 1961, 1963, 1965) tudi v tem primeru ne navajamo statistične analize ostankov izmišljenih ali z naključnim generatorjem ustvarjenih problemov. Smisel takšne analize bi bila primerjava s podobnimi metodami, ki pa jih v literaturi nismo zasledili.

5. ZAKLJUČEK

V članku je predlagana hevristična metoda optimizacije enodimenzionalnega razreza v primeru, ko je treba zaradi tehnoloških in logističnih vzrokov razdeliti naročilo na več manjših delov. Te dele nato optimiziramo posebej, vendar tako, da je skupna izguba materiala čim manjša. Predlagana hevristična metoda je implementirana v obliki računalniškega programa G-CUT. G-CUT omogoča oblikovanje skupin naročil in minimizacijo skupne izgube materiala. Uporaba programa je prikazana na preprostem primeru iz prakse. Predlagana metoda je primerna za reševanje problemov, kjer je razmerje med povprečno dolžino naročila in povprečno dolžino palice v skladišču sorazmerno majhno. V teh primerih je število palic iz skladišča, ki jih potrebujemo za izpolnitev naročila, le malo večje, kot je število kosov naročil. Na osnovi tega lahko domnevamo, da je večina praktičnih potreb po razdelitvi naročila na manjše dele povezana s tovrstnimi problemi. Pri večjih razmerjih pa bi bilo treba metodo razširiti z možnostjo oblikovanja podskupin, sestavljenih iz treh ali več naročil.

V nadaljevanju raziskovanja področja optimizacije razreza po skupinah bi bilo koristno nadomestiti minimizacijo izgube z minimizacijo stroškov tako kot na primer v (Erjavec, Trkman & Gradišar, 2008), pri čemer predmet optimizacije ne bi bila samo aktivnost razreza, ampak z njo povezane ostale aktivnosti, ki sestavljajo procese nabavljanja materiala (Trkman, McCormack, 2010), skladiščenja, logistike in proizvodnje.

LITERATURA

Alfieri, A., van de Velde, S. & Woeginger, G. (2004). Roll cutting in the curtain industry, or: A well-solvable allocation problem. *European Journal of Operational Research*, 183 (3), 1397–1404.

Alves, C. & de Carvalho, J. M. V. (2008). New integer programming formulations and an exact algorithm for the ordered cutting stock problem. *Journal of the Operational Research Society*, 59, 1520–1531.

- Belov, G. & Scheithauer, G. (2002). A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. *European Journal of Operational Research*, 141, 274–294.
- Bischoff, E. E. & Wäscher, G. (1995). Cutting and packing. *European Journal of Operational Research*, 84 (3), 503–505.
- Chauhan, S. S., Martel, A. & D'amour, S. (2008). Roll assortment optimization in a paper mill: An integer programming approach. *Computers & Operations Research*, 35 (2), 614–627.
- Cherri A. C., Arenales, M. N. & Yanasse, H. H. (2009). The one-dimensional cutting stock problem with usable leftover – A heuristic approach. *European Journal of Operational Research*, 196 (3), 897–908.
- Cui, Y., Gu, T. & Hu, W. (2009). A cutting-and-inventory control problem in the manufacturing industry of stainless steel wares. *Omega*, 37 (4), 864–875.
- Erjavec, J., Trkman, P. & Gradišar, M. (2008). Renovation of the cutting stock process. *International Journal of Production Research*, 47 (14), 3979–3996.
- Erjavec, J., Gradišar, M. & Trkman, P. (2010). Assessment of stock size to minimize cutting stock production costs. *International Journal of Production Economics*, 135 (1), 170–176.
- Gass, S. (1985). *Linear programming methods and applications*. New York: McGraw-Hill.
- Gilmore, P. C. & Gomory, R. E. (1961). A linear programming approach to the cutting stock-problem. *Operations Research*, 9 (4), 849–859.
- Gilmore, P. C. & Gomory, R. E. (1963). A linear programming approach to the cutting stock problem – Part II. *Operations Research*, 11 (6), 863–888.
- Gilmore, P. C. & Gomory, R. E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13 (1), 94–120.
- Gradišar, M., Erjavec, J. & Tomat, L. (2011). One-Dimensional Cutting Stock Optimization with Usable Leftover: A Case of Low Stock-to-Order Ratio. *International Journal of Decision Support System Technology*, 13 (1).
- Gradišar M., Jesenko J. & Resinovic G. (1997). Optimization of roll cutting in clothing industry. *Computers & Operations Research*, 24, 945–953.
- Gradišar M. et al. M. (1999). A sequential heuristic procedure for one-dimensional cutting. *European Journal of Operational Research*, 114 (3), 557–568.
- Gradišar M., Trkman P. (2005). A combined approach for solution of general one-dimensional cutting stock problem. *Computers & Operations Research*, 32 (7), 1793–807.
- Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management Science*, 6 (4), 366–422.
- Li, S. (1996). Multi-job cutting stock problem with due dates and release dates. *Journal of the Operational Research Society*, 47 (4), 490–510.
- Manrique, J. D., Al-Hussein, M., Bouferguene, A., Safouhi, H., & Nasser, R. (2011). Combinatorial Algorithm for Optimizing Wood Waste in Framing Designs. *Journal of Construction Engineering and Management*, 137 (3), 188–198.
- Matsumoto, K., Umetani, S. & Nagamochi, H. (2011). On the one-dimensional stock cutting problem in the paper tube industry. *Journal of Scheduling*, 14 (3), 281–290.

Trkman, P. & Gradišar, M. (2003). Optimization of the cutting stock process. *Journal of Mechanical Engineering*, 49 (9), 469–475.

Trkman, P. & Gradišar, M. (2007). One-dimensional cutting stock optimization in consecutive time periods. *European Journal of Operational Research*, 179 (2), 291–301.

Trkman, P. & McCormack, K. (2010). Estimating the benefits and risks of implementing e-procurement *IEEE Transactions on Engineering Management*, 57 (2), 338–349.

Venkateswarlu, P. (2001). The trim loss problem in wooden container manufacturing company. *Journal of Manufacturing Systems*, 20 (3), 166–176.