Računalne metode prikladne za rješavanje problema pakiranja pri uklapanju krojnih slika

Daniel Domović, mag. ing. comp.
Prof.dr.sc. **Tomislav Rolich**, dipl.ing.el.
Zavod za temeljne prirodne i tehničke znanosti
Sveučilište u Zagrebu Tekstilno-tehnološki fakultet
Zagreb, Hrvatska
e-mail: daniel.domovic@ttf.hr; tomislav.rolich@ttf.hr
Prispjelo 2.7.2015.

UDK 687.02:518
Pregled

U ovom je radu dan sustavan pregled problematike istraživanja dvodimenzionalnog problema računalnog uklapanja krojnih slika. Spomenuti problem i njegove podvrste osobito su proučavani u području računarske znanosti gdje su zajedničkim imenom opisani kao problem pakiranja. Problem pakiranja je u tome da se skup poligona mora umetnuti u veći spremnik (također poligon) uz uvjet da se poligoni u spremniku ne preklapaju, niti izlaze izvan granica spremnika. Svrha je minimiziranje površine slobodnog prostora između poligona, odnosno smanjenje površine spremnika. S obzirom na to da se problem pakiranja u praksi javlja u različitim industrijama, u radu je dan pregled podtipova problema pakiranja i njihova taksonomija, pregled metoda za detekciju preklapanja dvaju poligona: rasterska metoda, no-fit poligon, metoda izravne trigonometrije i D-funkcija, te grafovi ograničenja. Također, opisani su i neki od postojećih algoritama za rješavanje problema pakiranja.
Ključne riječi: *problem pakiranja, genetski algoritam, detekcija preklapanja, automatsko uklapanje krojnih slika, rasterska metoda, no-fit poligon, metoda izravne trigonometrije, D-funkcija, grafovi ograničenja*

1. Uvod

Unatoč krizi koju proživljava, odjevna industrija u Hrvatskoj i dalje je značajna po broju zaposlenih i činjenici da je riječ o industriji s velikim udjelom zaposlenih žena. Pritom je prepoznato je da napredak tekstilne i odjevne industrije leži u ulaganju u nove tehnologije, nove proizvode te specijalizaciji i izvozu [1].
Tehnologija proizvodnje u odjevnoj industriji duže se vrijeme nije značajno mijenjala. Radnik je i dalje važan dio proizvodnog procesa i specijaliziran je za obavljanje jedne ili više povezanih aktivnosti sa svrhom povezivanja tekstilnih dijelova u finalni proizvod [2].
Donedavno je razvoj odjevne industrije bio usmjeren samo na razvoj proizvodnje i povećanje njene učinkovitosti s naglaskom na nisku cijenu konačnog proizvoda. Danas se mnogo pažnje usmjerava i na tzv. čistu proizvodnju koja podrazumijeva optimalno korištenje resursa duž cijelog životnog ciklusa proizvoda u proizvodnom procesu, kako bi se utjecaj na okoliš što više smanjio [3]. Odjevna industrija također se mora prilagoditi novim tržišnim zahtjevima kojima pripada i brza i kvalitetna izrada individualno prilagođenih odjevnih predmeta.
S tom motivacijom, u ovom radu bit će riječi o segmentu optimizacije proizvodnje odjevnog predmeta sa svrhom optimiranja utroška materijala pri uklapanju krojnih slika. U strukturi varijabilnih troškova odjevnog predmeta udio troškova materijala iznosi 60 - 70 %, dok je udio izrade 30 - 40 %. Stoga je u tehnološkoj fazi krojenja potrebno izvršiti optimalno uklapanje krojnih dijelova kako bi se što bolje iskoristio krojni materijal. Time se smanjuju troškovi

proizvodnje i povoljno regulira zbrinjavanje otpada od iskrojavanja [4]. U računarskoj znanosti ovaj problem dio je šireg skupa problema, koji se općenitijim pojmom naziva problemom pakiranja. *Problem pakiranja takav je optimizacijski problem u kojem se više manjih elemenata mora, bez međusobnog preklapanja, rasporediti unutar granica spremnika.*

Praktična primjena ove definicije svakodnevno se susreće u praksi, npr. pri pakiranju namirnica u dućanu u više vrećica. U odjevnoj industriji, spremnik iz prethodne definicije jest područje uklapanja krojne slike na kojeg se moraju optimalno rasporediti krojni dijelovi. Raspored krojnih dijelova mora biti takav da minimizira utrošak materijala, tj. smanji duljinu materijala i time poveća njegovo iskorištenje.

Problem pakiranja dugo intrigira znanstvenu i poduzetničku/industrijsku zajednicu jer se u praksi pojavljuje u brojnim oblicima i područjima istraživanja, a ima velik potencijal kojima bi se mogli smanjiti vrijeme i troškovi proizvodnje. S obzirom na to da se problem u praksi javlja u različitim granama industrije, istraživači su problem (imenom i specifičnim opisom) prilagodili svom istraživačkom okviru.

Podvrste problema pakiranja su [5]:

- problem rezanja (engl. *cutting stock problem*, industrijsko inženjerstvo/proizvodnja),
- problem optimiranja gubitka (engl. *trim loss problems*, operacijska istraživanja, menadžment, proizvodnja),
- problem pakiranja (*engl. bin packing*, kombinatorička optimizacija),
- problem naprtnjače (engl. *knapsack packing problem,* kombinatorička optimizacija),
- problem utovara (engl. *vehicle/pallet/container/car loading*),
- uklapanje krojnih slika (engl. *marker making/lay plan/strip-packing*, odjevna industrija),
- problem gniježđenja (engl. *nesting problem*, strojarstvo, brodogradnja).

U praksi postoje razvijene aplikacije koje rješavaju uklapanje krojnih slika. Iako detalje svojih algoritama vodeće svjetske kompanije nipošto ne otkrivaju, izazovno je proučiti koje sve računalne metode i algoritmi iz dostupne literature stoje iza poznatih računalnih programa za automatsko uklapanje krojnih slika, kao i za problem pakiranja uopće.

2. Opis problema pakiranja

U ovom poglavlju problem pakiranja bit će formalno definiran, a bit će opisani i različiti podtipovi problema pakiranja, kao i podjela podtipova problema pakiranja s obzirom na različite parametre poput dimenzionalnosti te veličine spremnika i elemenata.

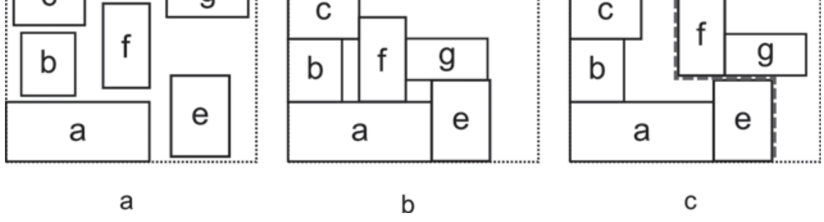
2.1. Definicija problema pakiranja

Zadan je skup elemenata (engl. *items*) $I = \{i_1, i_2, ..., i_n\}$, pri čemu je n broj elemenata skupa I, i spremnik (engl. *container*) C. Svi elementi $i \in I$ moraju se smjestiti u spremnik C, tako da je zadovoljen uvjet da se nijedna dva elementa i_i i i_j međusobno ne preklapaju (vrijedi: $i_i \cap i_j = 0$) i da nijedan element ne izlazi izvan granica spremnika (vrijedi: $i_i \cap C = i_i$) [6].

Ovom definicijom definirana su dva skupa elemenata i njihov razmještaj:

- Spremnik C: u dvije dimenzije spremnik je poligon pravilnog (najčešće pravokutnog) ili nepravilnog oblika (konveksni ili konkavni poligon) u koji je potrebno rasporediti elemente iz skupa I koji ne smiju izlaziti izvan granica spremnika. Za pravokutni spremnik C definira se širina i visina spremnika. Za spremnik u tri dimenzije definira se i dubina spremnika. U kontekstu računalnog uklapanja krojnih slika, spremnik je materijal pravokutnog oblika, fiksne širine i proizvoljne duljine.
- Skup elemenata I: skup od n pravilnih ili nepravilnih poligona iz skupa I koji se trebaju smjestiti u jedan ili više spremnika C (ovisno o tipu problema). Rotacije elemenata mogu, ali i ne moraju biti dozvoljene prilikom raspoređivanja unutar spremnika. U kontekstu računalnog uklapanja krojnih slika, elementi podrazumijevaju krojne dijelove koji su, zapravo, poligoni najčešće nepravilnog oblika.
- Razmještaj (engl. *placement*) P: pozicija elemenata u spremniku. Kako su elementi iz skupa I definirani koordinatama i referentnom točkom, razmještaj poligona definiran je pozicijom referentne točke u spremniku. U kontekstu računalnog uklapanja krojnih slika, krojni dijelovi moraju biti razmješteni na materijalu tako da se optimira duljina materijala u svrhu povećavanja iskorištenja materijala.

Poligoni (sl.1a) se mogu razmjestiti na način da im je razmještaj normaliziran (sl.1b) ili polu-normaliziran (sl.1c). Normalizirani razmještaj poligona onaj je razmještaj u kojem se nijedan poligon više ne može pomaknuti lijevo niti dolje a da ne uzrokuje preklapanje. Pri polu-normaliziranom razmještaju svaki poligon



Sl.1 Razmještaj poligona: a) poligoni, b) normalizirani razmještaj, c) polu-normalizirani razmještaj

stavljen je krajnje lijevo i krajnje dolje (bez preklapanja) u skladu s trenutnom konturom pakiranja.

Ranije postavljena definicija problema pakiranja može se dodatno poopćiti prema Wäscheru – potrebno je odabrati sve ili dio manjih elementa, grupirati ih u jedan ili više podskupova i svaki od podskupova dodijeliti jednom velikom objektu tako da su zadovoljeni sljedeći geometrijski uvjeti: svi mali elementi moraju biti smješteni unutar granica velikog objekta i mali elementi ne smiju se međusobno preklapati, na način koji optimira (jednodimenzionalnu ili višedimenzionalnu) funkciju cilja. Ovisno o tipu problema, rješenje se može dobiti korištenjem dijela ili svih velikih objekata, te dijela ili svih malih elemenata [7].

Ovom apstraktnijom definicijom postavljena je generalizacija koja odgovara široj skupini problema. U ovom radu fokus je na tri tipa problema: minimizaciji broja spremnika (napraviti razmještaj na način da se upotrijebi što manji broj spremnika), problemu odabira podskupa (odabrati najbolji podskup elemenata za razmještaj u spremnik u skladu s funkcijom cilja), te minimizaciji veličine spremnika (kao pri uklapanju krojnih slika) [6].

2.2. Primjeri problema pakiranja

2.2.1. Jednodimenzionalni problemi

Jednodimenzionalni problem rezanja (engl. *cutting-stock problem*) problem je koji se javlja u industriji tekstila, papira i metala u kojima se materijali pohranjuju u obliku smotaka određenih dimenzija. Ako je naručitelj zatražio isporuku više manjih smotaka materijala različitih dimenzija, nastojanje dobavljača je iz što manje velikih smotaka materijala izrezati dijelove prema mjerama koje želi naručitelj. Svrha je smanjiti količinu otpada koja se dobije rezanjem materijala.

Povezani problem je *jednodimenzionalni problem pakiranja* (engl. *one-dimensional bin-packing problem*) kojem je svrha pronaći minimalni broj spremnika u koje bi se moglo smjestiti sve elemente skupa *I*.

Iako su u suštini ova dva problema identična (dovoljno je zamijeniti pojam velikog smotka s pojmom kutije i malog smotka s elementom), autori ih razlikuju prema strukturi manjih elemenata. Problem rezanja koristi se kada su dimenzije manjih smotaka materijala relativno homogene (ima ih svega nekoliko različitih), dok se problem pakiranja koristi kada su elementi heterogeni što podrazumijeva veći broj elemenata različitih dimenzija.

Oba problema pripadaju nadskupini problema *minimizacije broja spremnika* s obzirom na to da je svrha minimizirati broj velikih smotaka ili broj kutija.

Učestali problem pakiranja je i jednodimenzionalni problem naprtnjače (engl. *one-dimensional knapsack problem*). Na raspolaganju je naprtnjača kapaciteta *C* i *n* elemenata iz skupa *I* kojima je pridijeljena težina *w* i cijena *p*. Potrebno je odabrati takav podskup elemenata iz skupa *I* koji nakon pohrane u naprtnjaču donose najveći prihod, pri čemu njihova težina ne smije biti veća od kapaciteta naprtnjače *W* (1). Vrijednost x_i označava pripadnost elementa *i* odabranom podskupu.

$$\max \sum_{i\in I} p_i x_i \text{n}$$
$$t.d. \quad \sum_{i\in I} w_i x_i \leq W$$
$$x_i \in \{0,1\}, i \in I \qquad (1)$$

Jednodimenzionalni problem naprtnjače pripada nadskupini problema *odabira podskupa* [6, 8].

2.2.2. Višedimenzionalni problemi odabira podskupa

Dvodimenzionalni problem naprtnjače (engl. *two-dimensional knapsack problem*) sličan je svojoj jednodimenzionalnoj inačici, s razlikom da je spremnik (naprtnjača) u ovom slučaju ploča dimenzija $W\times H$, a zadan je skup elemenata *I* u kojem svaki dvodimenzionalni element ima određenu cijenu p_i. Svrha je pronaći podskup elemenata *I'* koji donosi maksimalni prihod, odnosno podskup koji maksimizira vrijednost ($\sum_{i\in I} p_i x_i$ (oznaka x_i označava pripada li element *i* odabranom podskupu elemenata: $x_i \in \{0,1\}$, $i \in I$), a može se staviti u granice spremnika. Ovaj se problem u literaturi spominje i kao *dvodimenzionalni problem rezanja* (engl. *stock size cutting stock problem*).

Povezani problem je i *problem pakiranja paleta* (engl. *pallet loading problem*) kojem je svrha pronaći optimalni razmještaj identičnih objekata (kutija) koji se moraju staviti na paletu. Iako je ovdje riječ o 3D problemu, njegovom se rješavanju pristupa na način da se najprije kutije rasporede na samo jedan sloj koji se potom multiplicira.

2D problem naprtnjače moguće je generalizirati u tri dimenzije. Posebna takva podvrsta problema je *problem utovara* (engl. *container loading problem*, CLP) koji se pojavljuje u transportnoj industriji poglavito u transportu brodskim spremnicima. Ovaj se problem može poistovjetiti sa 2D problemom naprtnjače, pri čemu je cijena elementa jednaka njegovom volumenu. Razlika između ova dva problema je u tome što CLP daje bolju učinkovitost kada se koriste veliki i izduženi spremnici. Uz to, CLP se uglavnom koristi za pakiranje velikog broja kutija čija je suma volumena približno jednaka volumenu spremnika. Također, svrha je pronaći takvo pakiranje elemenata koje bi maksimiziralo volumen koji ti elementi zauzimaju. Pritom treba imati na umu stabilnost elemenata – ne smiju „ostati u zraku“ i ne smiju pasti jednom kada se spremnik pokrene [6].

2.2.3. Višedimenzionalni problem minimiziranja broja spremnika

Jednodimenzionalni problem pakiranja može se generalizirati na više dimenzija sa svrhom minimizacije broja spremnika u koje je potrebno spremiti skup elemenata. U literaturi su najčešće opisani pravokutni elementi kojima nije dozvoljeno rotiranje.

Inačica višedimenzionalnog problema minimiziranja broja spremnika jest problem utovara više spremnika (engl. *multi-container loading problem*, MCLP) i problem utovara na više paleta (engl. *multi-pallet loading problem*, MPLP).

Kod MPLP-a pakiraju se elementi različitih dimenzija sa svrhom pronalaska minimalnog broja paleta potrebnih za utovar. MPLP se razlikuje od MCLP-a u tome da je spremnik manji (palete) i postoje stroži uvjeti za stabilnost [6].

2.2.4. Minimizacija veličine spremnika

Problem minimizacije veličine spremnika javlja se u dvije ili više dimenzija kada je potrebno pronaći minimalnu veličinu spremnika koji bi bio dovoljno velik da se u njega spremi zadani podskup elemenata. Rješavanje problema može biti različito: npr. minimizacija samo jedne dimenzije spremnika (npr. problem gniježđenja kod kojeg se minimizira duljina materijala, a širina je fiksna), minimizacija površine/volumena spremnika ili minimiziranje polumjera kružnog spremnika.

Problem raspoređivanja (engl. *strip-packing problem*) je problem koji svoju primjenu nalazi u odjevnoj industriji pri čemu se krojni dijelovi moraju rasporediti u najbolji mogući raspored na materijalu kako bi se postigla njegova maksimalna iskoristivost. Općenitije, elementi skupa I moraju se rasporediti u pravokutni spremnik pri čemu je jedna dimenzija spremnika poznata (svaki tekstilni smotak ima definiranu širinu materijala), dok se druga dimenzija spremnika (duljina) mora minimizirati. Elementi (krojni dijelovi) su nepravilnog oblika. Takav oblik problema često se naziva i *problem gniježđenja* (engl. *nesting problem*).

Problem minimizacije površine (engl. *area minimization*) pojavljuje se kada treba optimirati više od jedne dimenzije spremnika. Primjer takvog problema u dvije dimenzije je VLSI (engl. *very large scale integrated circuit*) problem razmještaja pri čemu je potrebno pronaći minimalnu površinu pravokutne pločice na koju bi se mogao postaviti zadani skup pravokutnih modula.

Pakiranje kružnica (engl. *circle packing*) je zadatak kod koga je potrebno pronaći kružnicu minimalnog polumjera koja može opisati zadani skup kružnica. Najčešće se pojavljuje u industriji kablova ili uljnih cjevovoda [6].

2.3. Taksonomija problema pakiranja

S obzirom na općenitost Wäscherove definicije problema pakiranja i činjenice da istraživači dolaze iz različitih područja, često se isti tip problema pojavljuje pod različitim imenima (kao što je to vidljivo iz uvoda). Iz tog razloga predložene su različite taksonomije temeljene na skupovima osnovnih tipova problema [9].

Prvu taksonomiju problema pakiranja predložio je Dyckhoff. Prema njegovoj klasifikaciji problemi se dijele prema četiri osnovna kriterija: dimenzionalnost, svrha zadatka, veliki objekti i mali objekti. Prema dimenzionalnosti problemi mogu biti 1D, 2D, 3D ili n-dimenzionalni, pri čemu je n veći od 3. Prema tipu zadatka problemi se dijele na one kojima je cilj rasporediti elemente kako bi se minimizirao ili broj spremnika ili veličina jednog spremnika te time

Tab.1 Taksonomija problema pakiranja prema Wäscheru

Problemi maksimizacije izlaza		**Problemi minizacije ulaza**	
IPPP	(engl. *identical item packing problem*) – problem pakiranja paleta	SBSBSP	(engl. *single bin size bin packing problem*) – problem pakiranja
SKP	(engl. *single knapsack problem*) – problem naprtnjače i problem utovara spremnika	MBSBPP	(engl. *multiple bin size bin packing problem*) – problem pakiranja s različitim veličinama kutija
MIKP	(engl. *multiple identical knapsack problem*) – svrha je odabrati podskup elemenata za više istih naprtnjača; problem utovara više spremnika i problem utovara više paleta	SSSCSP	(engl. *single stock size cutting stock problem*) – problem pakiranja kutija sa slabo heterogenim elementima
MHKP	(engl, *multiple heterogenous knapsack problem*) – inačica MIKP-a s različitim naprtnjačama	RBPP	(engl. *residual bin packing problem*) – problem pakiranja kutija sa strogo heterogenim tipovima kutija
SLOPP	(engl. *single large object placement problem*) – oblik slabo heterogenog SKP-a	MSSCSP	(engl. *multiple stock size cutting stock problem*) – inačica SSSCSP-a s različitim tipovima kutija
MILOPP	(engl. *multiple identical large object placement problem*) – inačica SLOPP-a s više identičnih naprtnjača	ODP	(engl. *open dimension problem*) – problemi minimizacije dimenzije spremnika (npr. problem raspoređivanja) i problemi minimizacije površine
MHLOPP	(engl. *multiple heterogenous large object placement problem*) – inačica MILOPP-a s različitim naprtnjačama	RCSP	(engl. *residual cutting stock problem*) – MSSCSP sa strogo heterogenim tipovima kutija.

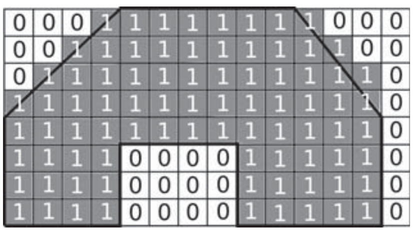
smanjio „otpad“ (prema Wäscheru: minimizacija ulaza) i na probleme kojima je svrha naći što bolji podskup elemenata kako bi se maksimizirao prihod tih elemenata (prema Wäscheru: maksimizacija izlaza). Veliki su objekti spremnici kojih može biti: jedan, više različitih ili više identičnih i u njih se spremaju mali objekti. Mali su objekti elementi podijeljeni prema broju i stupnju homogenosti [6].

Wäscher je revidirao Dyckhoffovu taksonomiju zadržavši kriterije dimenzionalnosti i svrhe zadatka, a revidirao je kategorije velikih i malih objekata, te dodao novu kategoriju: oblik malih objekata [6, 7].

Sličnost elemenata klasificira se u tri skupine: identični, slabo heterogeni i strogo heterogeni elementi. Skup velikih objekata također se klasificira u tri skupine: jedan objekt (problem maksimizacije izlaza), skup identičnih objekata ili skup heterogenih objekata. Za minimizaciju ulaza skup velikih objekata može se dodatno klasificirati kao: identični ili heterogeni.

Novom kategorijom definira se oblik malih elemenata koji mogu biti: pravilni (pravokutni, kružni ili cilindrični elementi) ili nepravilni [7].

Svojom taksonomijom Wäscher je predložio i novu nomenklaturu problema (tab.1). Ime problema sastavlja se navođenjem dimenzionalnosti problema, potom oblikom ulaznih objekata i definiranjem tipa identificiranog problema. Tako je 2-dimenzionalni pravokutni SKP (engl. *single knapsack problem*) dvodimenzionalni problem naprtnjače s pravokutnicima [7].



Sl.2 Rasterska metoda prikaza poligona

3. Metode detekcije preklapanja objekata

Najvidljivija karakteristika problema pakiranja i jedna od prvih prepreka s kojom se istraživači susreću, je geometrija likova. Prema definiciji problema pakiranja uvjet za nalaženje dopuštenog razmještaja elemenata u spremniku je da se nijedna dva elementa u razmještaju ne preklapaju i da su svi elementi raspoređeni unutar granica spremnika. Dok je odgovor na pitanje da li se dva lika preklapaju, ne preklapaju ili dodiruju trivijalno ljudskom oku, izrada odgovarajućeg računalnog modela koji bi mogao opisati ovaj problem netrivijalan je zadatak i barijera je za istraživače. Odabir dobrog modela nije samo pitanje kvalitete rada modela već i složenosti implementacije.

Formalno definirano, neka je $s_i \subset \mathbb{R}^d$ skup koji se sastoji od i elementa u d-dimenzionalnom prostoru problema. Kako bi se osiguralo nepreklapanje dvaju elemenata, za svaki par elemenata s_i i s_j iz skupa S mora vrijediti $\mathrm{int}(s_i) \cap \mathrm{int}(s_j) = 0$, pri čemu je $\mathrm{int}(s)$ unutrašnjost elementa $s \in S$. Ovaj uvjet dozvoljava dodirivanje elemenata. Nadalje, neka spremnik zauzima prostor $C \subset \mathbb{R}^d$, pri čemu vrijedi $s_i \cap C = s_i$ za svaki element s_i [9].

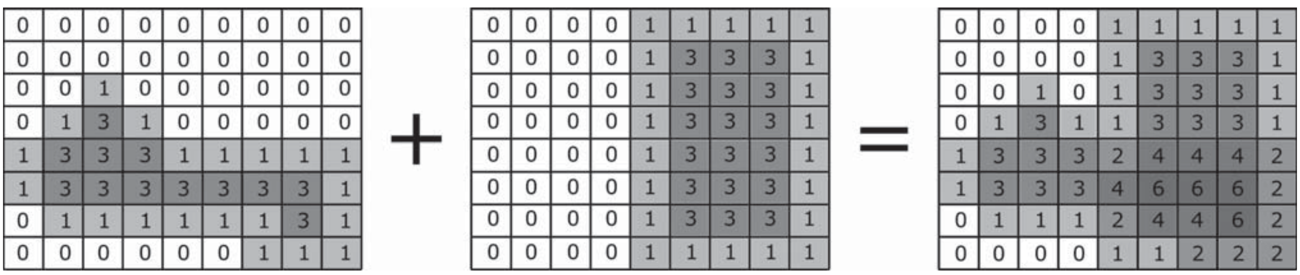
U nastavku rada opisane su najčešće metode detekcije preklapanja koje se pojavljuju u literaturi: rasterska metoda (engl. *raster method*), izravna trigonometrija (engl. *direct trigonometry*) i no-fit poligon (engl. *no-fit polygon*). Ove metode opisuju na koji se način elementi moraju postaviti jedan u odnosu prema drugom kako se ne bi preklapali. Opisana je i metoda grafa ograničenja (engl. *constraint graphs*) kojom se može izbjeći preklapajući razmještaj elemenata.

3.1. Rasterska metoda

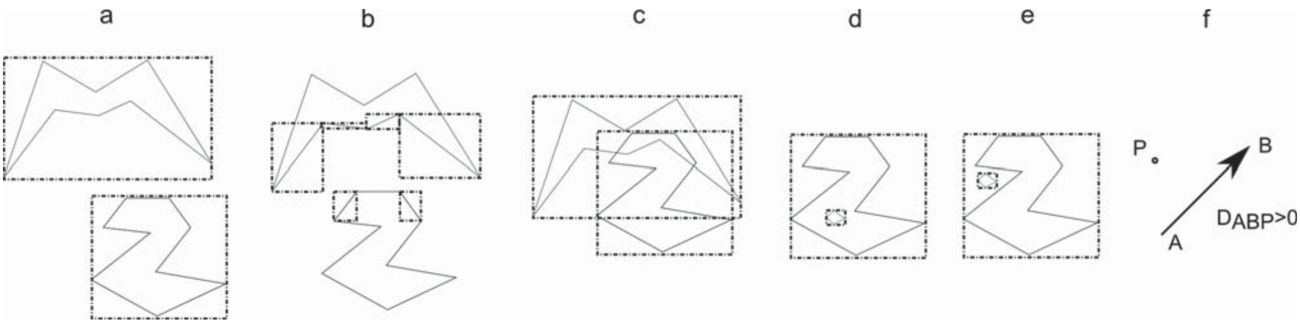
Ako je prostor spremnika kontinuiran, rasterskom metodom moguće ga je diskretizirati čime se dobiva mreža polja koju je moguće zapisati pomoću matrice.

Na sl.2 prikazana je najjednostavnija metoda kodiranja elemenata. U matricu se postavlja vrijednost 1 (ako se dio lika nalazi u rasterskom polju mreže) ili 0 (ako se lik ne nalazi u rasterskom polju mreže). Smještanje jednog objekta u spremnik odgovara zbrajanju matrice koja predstavlja razmještaj objekata i matrice koja opisuje objekt koji želimo umetnuti u razmještaj. Ako se pritom u nekom polju dobije vrijednost veća od 1, došlo je do preklapanja objekata [10].

Na sl.3 prikazan je oblik kodiranja matrice kojim je moguće detektirati dodirivanje objekata te njihovo preklapanje. Vrijednost 1 koristi se za kodiranje ruba objekta, a vrijednost 3 za vrijednost unutrašnjosti objekta. Ako se nakon zbrajanja matrice elementa s matricom razmještaja dobije vrijednost veća ili jednaka od četiri, razmještaj nije dozvoljen jer je riječ o preklapanju ruba i unutrašnjosti jednog objekta ili preklapanju unutrašnjosti dvaju elemenata. Vrijednosti manje od 2 označavaju da se



Sl.3 Metoda detekcije preklapanja rasterskom metodom



Sl.4 Ispitivanja preklapanja metodom izravne trigonometrije

dva elementa ne preklapaju. Vrijednost 2 označava dodirivanje dvaju elemenata [11].

Predložene su i neke složenije metode kodiranja [12]. Spremnik se može kodirati tako da mu je unutrašnjost označena znamenkama nula. Pikseli koji su na granici ili izvan spremnika kodiraju se na način da se u najdesniji piksel u retku matrice, koji je različit od nule, postavi vrijednost 1 i ta se vrijednost inkrementalno povećava za jedan s desna na lijevo. Objekti se kodiraju slično, ali je i granica objekta označena vrijednošću nula. Ovakvim načinom kodiranja u svakom je pikselu prikazana vrijednost koja opisuje udaljenost za koliko je potrebno pomaknuti objekt u desno kako bi se postigao dopušteni razmještaj. Tako se u jednom koraku može preskočiti više piksela. Kada se odredi optimalni razmještaj objekta u spremniku, objekt se kodira na isti način kao i spremnik [9, 10].

Prednost rasterskih metoda prikaza je u tome da se jednostavnim prebrojavanjem ćelija u željenom smjeru može izračunati udaljenost na koju je potrebno pomaknuti objekt kako bi se izbjegao nedopušteni razmještaj u kojem se poligoni preklapaju, kao i mjesto na koje je potrebno smjestiti dva poligona kako bi se dodirivali. Ove metode jednostavne su za kodiranje, te mogu opisati konveksne i konkavne poligone.

Nedostatak rasterske metode je memorijska zahtjevnost i nemogućnost preciznog prikaza poligona s neortogonalnim bridovima čime se smanjuje geometrijska informacija o obliku poligona. Taj se problem može ispraviti povećavanjem preciznosti rastera, ali time se povećavaju zahtjevi nad memorijom i vrijeme provjere preklapanja.

3.2. Izravna trigonometrija i *D*-funkcija

Metoda izravne trigonometrije koristi se kada je poligone potrebno precizno prikazati bez gubitka informacija o njihovom obliku. Količina informacija tada je proporcionalna broju vrhova poligona. Metoda izravne trigonometrije korisna je zbog toga što sadrži testove za sjecište linija i provjeru nalazi li se točka u poligonu. Za dobivanje te informacije, provode se do četiri testa (sl.4). Ovime je pretpostavljeno da su poligoni prikazani nizom vrhova i bridova [9].

U prvom se koraku ispituje preklapaju li se pravokutnici koji omeđuju dva poligona (sl.4a). Ako to nije slučaj, poligoni se ne preklapaju. U suprotnom, potrebno je ispitati da li se za svaki par bridova iz različitih poligona pravokutnici koji ih omeđuju preklapaju (sl.4b). Ako se nijedan takav par ne preklapa, ne preklapaju se ni poligoni. U suprotnom, za bridove kojima je otkriveno potencijalno preklapanje provodi se treći test. Trećim testom (sl.4c) ispituje se preklapanje para bridova, pri čemu je svaki brid iz drugog poligona. Ako je za sve parove bridova odgovor negativan, primjenjuje se i posljednji test, a u protivnom se poligoni preklapaju. Četvrtim testom ispituje se nalazi li se točka iz prvog poligona unutar drugog poligona i obrnuto. Ako to nije slučaj, poligoni se ne preklapaju (sl. 4e), u protivnom, došlo je do preklapanja poligona (sl.4d).

Analiza bridova u trećem testu obavlja se pomoću *D*-funkcije (2) koja daje relativnu poziciju točke *P* u odnosu na usmjereni brid *AB* i definira se kao:

$$D_{ABP} = ((X_A - X_B)(Y_A - Y_P) - (Y_A - Y_B)(X_A - X_P)= \quad (2)$$

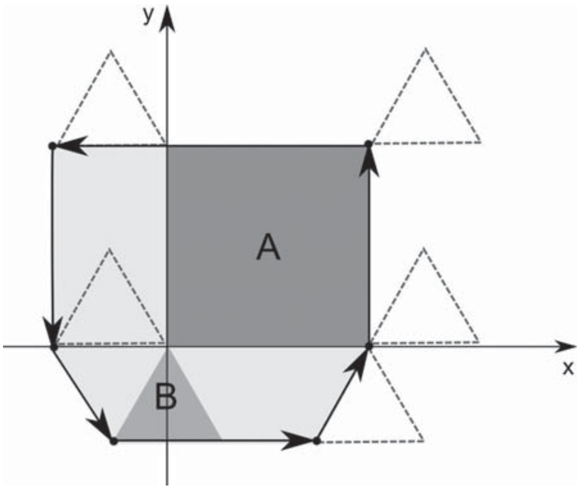
Ako je vrijednost $D_{ABP} > 0$, točka *P* nalazi se lijevo od brida *AB* (sl. 4f), ako je $D_{ABP} < 0$, točka *P* nalazi se desno od brida *AB*, a ako je $D_{ABP} = 0$, točka *P* nalazi se na bridu *AB* [13].

Prednost metode izravne trigonometrije jest mogućnost preciznog prikaza poligona, a njen je nedostatak nužnost provođenja mnogo računanja s decimalnim vrijednostima. Kod svake promjene pozicije poligona mora se ponovno provjeravati da li se poligoni preklapaju. Metoda *D*-funkcije najčešće se primjenjuje u iterativnim algoritmima za problem razmještaja.

3.3. No-fit poligon

No-fit poligon (engl. *no-fit polygon*, NFP) popularna je metoda za pronalaženje područja u kojem se dva poligona ne preklapaju. Ova je metoda brža od metode izravne trigonometrije, a omogućuje prikaz poligona pomoću bridova.

NFP_{AB} dvaju poligona *A* i *B* rezultantni je poligon dobiven operacijom klizanja poligona *B* oko fiksnog poligona *A*, pri čemu oba poligona imaju fiksnu orijentaciju (sl.5). Jedan vrh poligona *B* odabire se kao referentna točka. Staza koju opisuje referentna točka poligona *B* prilikom njegovog klizanja oko poligona *A* opisuje rub NFP-a. Prilikom klizanja poligoni *A* i *B* uvijek se moraju dodirivati, ali



Sl. 5. Konstrukcija *no-fit* poligona

nikada preklapati. U slučaju zamjene uloga poligona A i B, NFP_{BA} zarotiran je za 180° u odnosu na NFP_{AB} [9].

Unutrašnjost NFP-a sadrži sve pozicije u kojima se poligoni A i B preklapaju. Provjera da li se poligoni preklapaju obavlja se ispitivanjem odnosa referentne točke poligona B i NFP-a. Ako se referentna točka nalazi izvan granica NFP-a, poligoni A i B se ne preklapaju. Ako se referentna točka nalazi na granici NFP-a, poligoni A i B se dodiruju, a ako se referentna točka nalazi unutar NFP-a, dva se poligona preklapaju.

Rezultat se može proširiti i na pozicije ishodišta poligona A različite od (0, 0). Ako se A nalazi na poziciji (x, y), referentna točka poligona B mora se translatirati za ($-x$, $-y$) prije ispitivanja relativne pozicije u odnosu na NFP_{AB}. Tada se preklapanje otkriva provjerom da li je rezultat oduzimanja pozicije ishodišta poligona A i referentne točke poligona B unutar *NFP*-a.

Opisani NFP naziva se *outer-fit* NFP, a dodatno se razlikuje i *inner-fit* NFP. *Inner-fit* NFP nastaje klizanjem poligona po unutarnjoj strani spremnika čime se dobiva skup pozicija unutar spremnika u koje se može smjestiti klizajući poligon a da ne dođe do preklapanja [14].

Kretanje jednog poligona oko drugog modelira se, ranije opisanim, klizećim algoritmom. Kako bi se osiguralo da algoritam započne izvođenje u točki u kojoj se dva poligona ne preklapaju, točku s najvećom y-koordinatom klizećeg poligona B potrebno je smjestiti u točku s najmanjom y-koordinatom statičnog poligona A. Referentna točka poligona B definira prvi vrh NFP-a. Drugi vrhovi biraju se u smjeru obrnutom od kazaljke na satu nakon identificiranja para vrha i brida za sljedeći korak klizanja [11]. Tri su mogućnosti za klizanje poligona: vrh b_j klizi niz brid a_ia_{i+1}, brid b_jb_{j+1} klizi niz vrh a_i ili brid b_jb_{j+1} klizi niz brid a_ia_{i+1}. Pomoću D-funkcije bira se kombinacija vrha i brida klizanja.

Ako je jedan od poligona konkavan, postoji mogućnost da će klizanjem po klizećem bridu doći do preklapanja dvaju poligona. Za provjeru preklapanja i izračuna dopuštene udaljenosti do poligona kako ne bi došlo do preklapanja, svaki vrh iz poligona B projicira se na statični poligon A za udaljenost klizajućeg brida statičnog poligona. D-funkcija se koristi za provjeru preklapanja između projiciranog brida i bridova u statičnom poligonu A. Za bridove koji se preklapaju, minimalna udaljenost od originalne pozicije do točke preklapanja je maksimalna dozvoljena udaljenost za koju se B smije pomaknuti po klizećem bridu, a da ne dođe do preklapanja.

Za poligone koji mogu konstruirati NFP pomoću *klizećeg* algoritma kaže se da su jednostavno povezani, a takav je i rezultirajući NFP. Za poligone koji nemaju to svojstvo rezultirajući NFP može sadržavati rupe. Klizeći algoritam neće moći identificirati rupe, kao ni točke u kojima ne dolazi do preklapanja. Međutim, ovaj se nedostatak može ispraviti. Osnovna je pretpostavka da rubovi poligona koji nisu pređeni nakon provođenja klizećeg algoritma potencijalno predstavljaju rupu. Ako se moguće polazište za klizeći algoritam može pronaći na neposjećenim rubovima, onda se ekvivalentna klizeća operacija, u smjeru kazaljke na satu, može provesti kako bi se pronašle granica rupe (ili *inner-fit* poligon) [9].

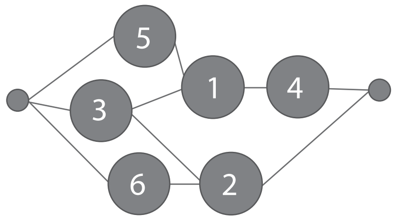
Osim klizećeg algoritma, NFP se može izračunati pomoću Minkowskijeve sume (engl. *Minkowski sum*) [15], te metodom dekompozicije elemenata u poligone u obliku zvijezde ili u konveksne poligone [16-19].

Iako je NFP dobar alat za detekciju odnosa između dvaju poligona, računanje NFP-a nije trivijalan zadatak. Zbog zahtjevnosti implementacije NFP-a istraživači ne koriste njegove prednosti. Sve spomenute metode imaju svoja ograničenja, npr. ne mogu identificirati pozicije za gniježđenje poligona koje bi rezultirale višestruko povezanim NFP-om ili imaju mnogo internih petlji, a mehanizmi za njihovu identifikaciju podliježu izravnom testiranju [9].

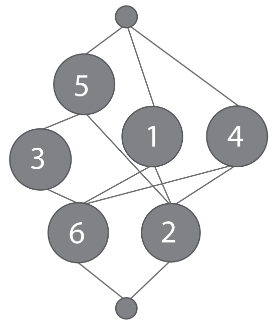
3.4. Graf ograničenja

Grafovi ograničenja skup su usmjerenih acikličkih grafova koji se koriste za opisivanje relativnog odnosa između pravokutnih elemenata [6]. Za d-dimenzionalan problem, d grafova moguće je iskoristiti za opisivanje pozicije svakog pravokutnog elementa – za svaku se dimenziju koristi jedan graf. Težine bridova u grafu koriste se za rekonstrukciju početne pozicije elementa.

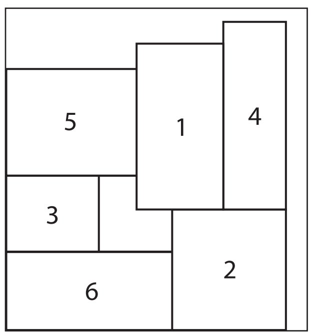
Na primjer, za dvodimenzionalni problem pravokutnici su opisani referentnom točkom (npr. koordinatom donjeg lijevog vrha), širinom i visinom. Pri modeliranju grafova za opis razmještaja elemenata koriste se dva grafa – G_h i G_v. Graf G_h opisuje raz-



Sl.6 Graf ograničenja u horizontalnom smjeru



Sl.7 Graf ograničenja u vertikalnom smjeru



Sl.8 Rekonstruirani razmještaj dobiven na temelju grafova ograničenja

mještaj elemenata u horizontalnom smjeru, dok graf G_v opisuje razmještaj elemenata u vertikalnom smjeru. U grafove se najprije dodaju ishodišni čvorovi *W* u graf G_h i čvor *S* u graf G_v. Potom se u svaki graf dodaje novi čvor koji predstavlja istoimeni pravokutnik i tako redom dok se u grafove ne postave čvorovi za svaki pravokutnik. Brid od čvora *a* do *b* u grafu G_h označava da se pravokutnik *a* nalazi lijevo od pravokutnika *b*, dok brid od *a* do *b* u grafu G_v označava da se pravokutnik *a* nalazi ispod pravokutnika *b*. U grafu G_h svakom se bridu pridjeljuje težina koja odgovara duljini pravokutnika *a*, dok u G_v ta težina odgovara širini pravokutnika *a*. Između svaka dva čvora koji se dodiruju smije postojati samo jedan brid. Težina brida od čvora *W* prema svim čvorovima koji se u rasporedu nalaze krajnje lijevo je 0. Isto vrijedi i za bridove od čvora *S* prema svim čvorovima koji se u rasporedu nalaze krajnje dolje.

Na sl.8 prikazan je primjer pakiranja pravokutnika dobiven na temelju grafova ograničenja u horizontalnom smjeru (sl.6) i grafa ograničenja u vertikalnom smjeru (sl.7). Težine bridova nisu označene zbog preglednosti, no odgovaraju duljini, odnosno širini pravokutnika iz G_h i G_v.

Grafovi ograničenja koriste se za rekonstrukciju pozicije svakog elementa u konačnom razmještaju. Pozicija pravokutnika u razmještaju označava se *x* i *y* koordinatom referentne točke i pronalazi se računanjem najduljeg puta od početnog čvora grafa do čvora koji predstavlja pravokutnik čiju poziciju se želi pronaći. Ukupna širina ili duljina razmještaja dobiva se pretraživanjem grafa.

Kada se za generiranje razmještaja koriste grafovi ograničenja ne može doći do preklapanja elemenata [6].

4. Pristupi rješavanju problema pakiranja

U ovom poglavlju navedene su karakteristične metode opisane u literaturi koje se koriste za rješavanje problema pakiranja.

4.1. Strategija dolje-lijevo

Strategija dolje-lijevo učestalo je korištena strategija razmještanja pravokutnika koja iskorištava prednost normaliziranog razmještaja pravokutnika.

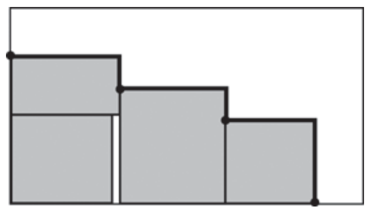
Neka je zadan skup pravokutnika *I* u poretku *L*. Strategijom dolje-lijevo, za svaki se pravokutnik $i \in I$, prema poretku iz *L* bira krajnje donja i krajnje lijeva moguća pozicija razmještaja [6].

Ovakav se razmještaj može postići u vremenskoj složenosti $O(n^2)$ [20]. Strategija dolje-lijevo često se koristi u kombinaciji s metaheurističkim algoritmima (npr. genetski algoritam) za rješavanje problema pakiranja pravokutnika. Pritom je razmještaj pravokutnika prikazan nizom *L* koji je ujedno i jedinka. Dakle svaka jedinka prikazuje jedan razmještaj [21, 22].

4.2. Metoda omotnica

Metoda omotnica (engl. *envelope*) ili profila (engl. *profiles*) korisna je prilikom pakiranja pravokutnika prema određenom redoslijedu, jedan po jedan [23]. Metoda omotnica koristi se za smanjivanje broja dohvatljivih (engl. *feasible*) pozicija razmještaja pravokutnika $i \in I$ i to rezanjem dijela dohvatljivog područja.

Novi se pravokutnik postavlja u spremnik tako da njegov donji lijevi vrh ne bude ispod i lijevo od bilo kojeg gornjeg desnog vrha prethodno razmještenih pravokutnika. Granica mogućih dohvatljivih pozicija stepeničasti je uzorak (omotnica, profil), a novi je pravokutnik moguće smjestiti samo tako da mu donji desni vrh dodiruje unutarnji vrh stepeničastog uzorka (točke na sl.9). Dodavanjem pravokutnika u razmještaj linija omotnice se širi, a broj dohvatljivih pozicija je konačan.



Sl.9 Grafički prikaz metode omotnice

Metoda omotnice može se koristiti u kombinaciji s drugim heuristikama poput strategije dolje-lijevo [23].

4.3. Metode apstraktnog prikaza

Razmještaj se može prikazati kao lista poligona pri čemu je svaki poligon prikazan pomoću koordinata svojih vrhova. Nedostatak ovakvog prikaza razmještaja je taj što se pomoću njega može prikazati i razmještaj s preklapanjima, kao i taj da je translaciju iz jednog nepreklapajućeg razmještaja u drugi zahtjevnije izvesti zbog translacije velikog broja vrhova. Mnoge heuristike koriste neki oblik apstraktnog prikaza razmještaja elemenata kako bi izbjegle nedostatke izravnog prikaza.

Apstraktnim prikazom želi se prikazati razmještaj opisivanjem odnosa između elemenata u razmještaju. Apstraktni prikaz razmještaja potom se podvrgava algoritmu dekodiranja kako bi se dobila rekonstrukcija razmještaja [6].

Jedan od primjera apstraktnog razmještaja su parovi nizova (engl. *sequence pair*) [24]. Par nizova je par lista koje se sastoje od permutacija indeksa n pravokutnika $\{1, 2, \ldots, n\}$ (npr. za $n = 5$, $a = <2, 1, 4, 5, 3>$, $b = <2, 5, 1, 4, 3>$). Odnos između pravokutnika dekodira se prema relaciji:

$$(<\ldots m_i \ldots m_j \ldots>, <\ldots m_i \ldots m_j \ldots>) \rightarrow \rightarrow m_i \text{ je lijevo od } m_j, \quad (3)$$

$$(<\ldots m_j \ldots m_i \ldots>, <\ldots m_i \ldots m_j \ldots>) \rightarrow \rightarrow m_i \text{ je ispod } m_j. \quad (4)$$

Par nizova predstavlja pakiranje elemenata čija se veličina može izračunati metodom detekcije najduljeg zajedničkog podniza. Metodom traženja najduljeg podniza pronalaze se dimenzije razmještaja, a ako je svaki pravokutnik definiran referentnom točkom, širinom i duljinom, moguće je rekonstruirati i koordinatu referentne točke pravokutnika, a time i poziciju pravokutnika u razmještaju [25]. Parovi nizova koriste se za modeliranje grafova ograničenja. Na sl.8 prikazan je razmještaj pravokutnika za par nizova (<5, 3, 1, 4, 6, 2>, <6, 3, 2, 5, 1, 4>). Grafovi ograničenja često se koriste u kombinaciji s heurističkim metodama poput simuliranog kaljenja koje upravlja lokalnom pretragom pri izmjeni redoslijeda permutacija u nizovima [26]. Ova metoda uspješno se koristi za razmještanje pravokutnika u pravokutni spremnik s postotkom iskoristivosti prostora većim od 90 %.

Algoritam koji koristi grafove ograničenja za rješavanje problema pakiranja složenosti je $O(n^2)$ [24]. Poboljšanje je dano algoritmom FAST-SP pomoću kojeg se razmještaj traži izračunavanjem najduljeg zajedničkog podniza u paru nizova, i to u $O(n \cdot \log[\log(n)])$ složenosti [27]. Korištenjem ovog pristupa nije potrebno konstruirati horizontalni i vertikalni graf ograničenja i tada računati najdulji podniz u grafu. Kao posljedica toga, algoritam FAST-SP može ispitati više parova nizova i dati bolji razmještaj u kraćem izvođenju.

Neki autori ne iskorištavaju prednosti grafova ograničenja, već parove nizova interpretiraju kao redoslijed slaganja, pri čemu su sva pakiranja seminormalizirana, tj. svaki poligon postavljen je maksimalno lijevo i dolje u skladu s trenutnom konturom za pakiranje. Ti algoritmi mogu biti složenosti $O(n^2)$, ali i $O(n \cdot \log(n))$ [28].

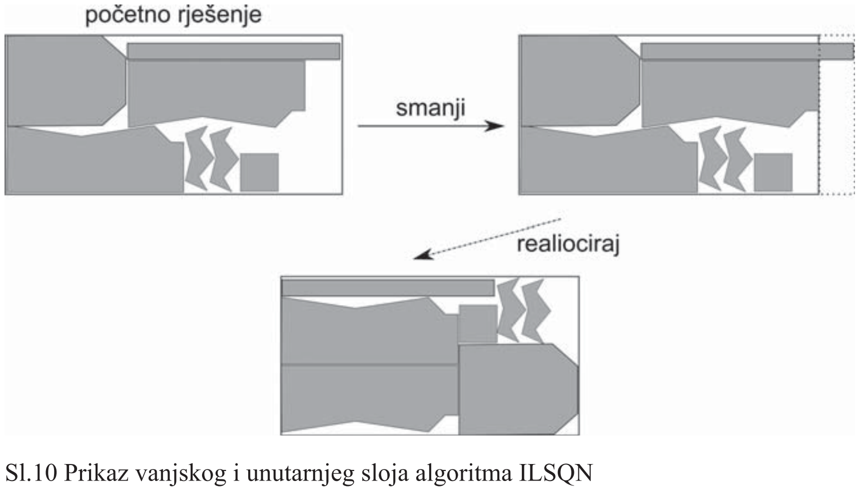
4.4. Metode temeljene na nelinearnom programiranju

Pri rješavanju problema uklapanja krojnih slika koriste se i metode zasnovane na nelinearnom programiranju poput *Iterated Local Search Quasi-Newton* (ILSQN) [29]. Algoritam ILSQN koristi kvazi-Newtonovu metodu za rješavanje optimizacijskog problema bez ograničenja u iterativnoj lokalnoj pretrazi. Svrha lokalne pretrage je minimizirati količinu preklapanja poligona i izbjeći postavljanje elemenata izvan granica spremnika.

Algoritam ILSQN, pri fiksnoj širini spremnika, raspoređuje poligone po pravokutnom spremniku bez međusobnog dodirivanja. Početna duljina spremnika L određena je kao udaljenost između ruba krajnje lijevog i krajnje desnog poligona. Tijekom izvođenja algoritma do zadanog vremenskog ograničenja izmjenjuje se rad dvaju slojeva: vanjskog i unutarnjeg (sl.10).

Vanjski sloj traži optimalnu duljinu spremnika njenim iterativnim smanjivanjem i povećanjem. Nakon ove radnje trenutni raspored poligona može postati nedohvatljiv, npr. poligoni mogu izlaziti izvan granica spremnika. Nedohvatljive slučajeve poboljšava rad unutarnjeg sloja stvarajući optimalna lokalna rješenja pozivom funkcije za minimizaciju preklapanja. Unutarnji sloj iterativni je algoritam lokalnog pretraživanja koji se sastoji od dvije metode – zamjena dvaju poligona i razmicanje poligona koji se preklapaju. Kod zamjene poligona biraju se dva nasumična poligona kojima se pri zadanoj duljini spremnika i fiksnom rasporedu drugih poligona nastoji pronaći nova pozicija u rasporedu s najmanjim preklapanjem. Metoda razmicanja poligona potom simultano djeluje na sve poligone u rasporedu primjenjujući kvazi-Newtonovu metodu za dobivanje optimalnog rasporeda s minimalnim preklapanjem u tom koraku.

Prednost algoritma ILSQN je u tome što vodi računa o rotaciji poligona [29, 30].



Sl.10 Prikaz vanjskog i unutarnjeg sloja algoritma ILSQN

4.5. Algoritmi zasnovani na evolucijskom računanju

Problem pakiranja može se uspješno riješiti primjenom algoritama zasnovanih na evolucijskom računanju. U literaturi je opisan pristup rješavanja pomoću genetskog algoritma (engl. *Genetic Algorithm, GA*) [31], algoritma optimizacije rojem čestica (engl. *Particle Swarm Optimization, PSO*) [32] i algoritma optimizacije kolonijom mrava (engl. *Ant Colony Optimization*, *ACO*) [33].

Genetski algoritam metoda je za rješavanje problema pretraživanja i optimizacije zasnovana na evolucijskim principima iz prirode. Sastavni dio algoritma su kromosomi – jedinke koje prikazuju rješenje i čine populaciju jedinki. GA se sastoji od iterativne procedure u kojoj se jedinke mogu reproducirati i time stvoriti nove jedinke i novu populaciju. Reproducirati se mogu sve jedinke, ali bolje će jedinke imati veću vjerojatnost da za taj proces budu izabrane. Primjerenost (dobrota) jedinke određuje se evaluacijom funkcije cilja. Nove jedinke nastaju križanjem čime preuzimaju dobra svojstva svojih roditelja. Lošije jedinke s vremenom nestaju iz populacije. Neke od novih jedinki mogu mutirati nakon procesa križanja.

Genetski algoritam daje dobra rješenja za problem pakiranja u kombinaciji s nekim algoritmom lokalne pretrage. Pri tome se GA koristi za određivanje rasporeda poligona (jedinka u GA je permutacija poligona), dok se lokalna pretraga koristi za traženje razmještaja poligona.

Algoritam lokalnog pretraživanja poput kvazi-Newtonove metode koristi se za rekonstrukciju optimalnog razmještaja tako da se minimizira količina „otpada“ što ujedno odgovara funkciji cilja kod genetskog algoritma. Funkcija cilja koja se optimira odgovara površini nezauzetog prostora spremnika kada se u njega razmjeste svi poligoni [34].

U hibridizaciji s GA moguće je primijeniti i algoritam za rješavanje dvodimenzionalnog problema gniježđenja nepravilnih poligona temeljen na algoritmu NFP i na načelu najnižeg centra gravitacije (engl. *lowest gravity center*, LGC) [31]. Prednost ovog algoritma je u tome što uspješno generira razmještaj poligona kod omogućene rotacije i uspješno rješava problem umetanja poligona u rupe nastale nakon uklapanja dvaju poligona. S obzirom na to da je prostor pretraživanja beskonačan jer svaka rotacija ili translacija bilo kojeg poligona dovodi do novog rasporeda, autori nastoje smanjiti prostor pretraživanja raspoređivanjem poligona u točke NFP-a na temelju minimalnog gravitacijskog centra poligona. Postupak razmještanja poligona u raspored započinje izračunavanjem gravitacijskog centra poligona i odabirom referentne točke poligona. Potom se izračunava *inner-fit* inačica NFP-a između poligona i spremnika. Poligon se smješta na poziciju s najnižim centrom gravitacije, pri čemu se njegova referentna točka mora nalaziti na rubu NFP-a (poligon time dodiruje rub spremnika). Na toj poziciji poligon se rotira. Kako se nakon rotacije poligona dobiva drugačija pozicija gravitacijskog centra, za konačni odabir smještaja poligona odabire se onaj poligon koji nakon rotacije ima minimalni centar gravitacije. Centar gravitacije računa se dijeljenjem poligona na trake (bridovi poligona projiciraju se na x-os, a druge dvije stranice paralelne su s y-osi) i računanjem gravitacijskog centra za svaku traku, nakon čega se računa ukupni centar gravitacije poligona. Budući da se poligoni u raspored gnijezde jedan po jedan, kako bi se dobilo što bolje iskorištenje spremnika ključno je pronaći najbolji redoslijed poligona. Kada bi se dijelovi sortirali po veličini i potom raspoređivali u spremnik, moguće je stvaranje velikih praznina između poligona i spremnika koju bi mogao popuniti neki manji poligon. Pristup rekurzivnog popunjavanja rupa (engl. *hole recursive nesting approach*) rješava ovaj problem.

Poligoni se sortiraju po veličini, od najvećeg prema najmanjem i tim se redom gnijezde. Pri gniježđenju poligona mogu nastati praznine u koje se može smjestiti jedan ili više manjih poligona. Kako bi se maksimizirala iskoristivost praznina, rekurzivno se ispituje pristaje li neki od manjih poligona u nastalu prazninu. Ako pristaje, mijenja se redoslijed gniježđenja tako da se manji poligon gnijezdi prije poligona koji je uzrokovao prazninu.

Ovako dobivena rješenja mogu se iskoristiti kao početne vrijednosti za GA. Za određivanje redoslijeda poligona autori koriste *order-based* genetski algoritam, pri čemu je kromosom permutacija koja predstavlja poredak poligona. Prva je jedinka raspored poligona poredanih po veličini, a ostale se jedinke dobivaju mutacijom početne. U procesu selekcije jedinki koristi se proporcionalan izbor, a kao operator križanja koristi se *order-based* crossover [20]. Prilikom mutacije, dva gena u kromosomu zamijene mjesto u poretku (s vjerojatnošću mutacije između 0,1 i 0,2). Ovaj algoritam postiže bolje rezultate u minimizaciji visine spremnika, postotku iskoristivosti spremnika i računalnom vremenu u usporedbi s algoritmom u [35].

Genetski algoritam može se primijeniti i u kombinaciji s heurističkim metodama poput strategije dolje-lijevo [36]. Pomoću GA pretražuje se prostor rješenja, a za dekodiranje rješenja koriste se strategija dolje-lijevo (engl. *bottom-left*, BL) i strategija dolje-lijevo-ispuni (engl. *bottom-left-fill*, BLF) koja ujedno može popuniti praznine koje nastaju strategijom BL i to alokacijom najniže dovoljno velike regije. Dekodiranjem rješenja dobiva se raspored elemenata. Rješenje (najbolja jedinka) je niz cjelobrojnih vrijednosti koje određuju redoslijed. Za kodiranje jedinke koristi se kodiranje temeljeno na poretku (engl. *order-based encoding*). Pri ovakvom kodiranju prikladno je koristiti *partially matched crossover* (PMX) i *order-based* mutaciju kako bi se postigli dohvatljivi rasporedi. BLF strategija daje bolja rješenja od BL strategije razmještanja poligona.

4.6. Odabrani heuristički algoritmi

Pored ranije opisanih algoritama, u optimizacijskom procesu može se koristiti i algoritam simuliranog kaljenja (engl. *simulated annealing*). Tada se za prikaz razmještaja koristi uređena lista elemenata, a neka se heuristička metoda koristi za dekodiranje razmještaja [14].

Kako bi se osiguralo stvaranje dohvatljivih rasporeda koristi se koncept regije bez kolizija (engl. *collision free region*). Regija bez kolizija područje je unutar spremnika u koji se može smjestiti poligon bez da dođe do preklapanja s prethodno razmještenim poligonima.

S obzirom da su dosadašnji algoritmi za giljotinsko pakiranje bili primjenjivani samo na pravokutnicima, istraživanju giljotinskog pakiranja pridonio je algoritam za giljotinsko pakiranje konveksnih poligona [37].

Pri pakiranju poligona nepravilnih oblika, autori ih često aproksimiraju najmanjim opisanim pravokutnikom i tada koriste neku od metoda za pakiranje pravokutnika. Nedostatak ovog pristupa je velika količina otpada između rubova opisanog pravokutnika i aproksimiranog poligona.

5. Zaključak

U radu je sustavno prestupljeno problemu pakiranja, pri čemu je dat pregled radova promatrane problematike. Rezultat istraživanja su identificirana područja za budući rad i daljnje istraživanje. Prema literaturi, u 2D pakiranju, najviše se istraživalo područje razmještaja pravokutnika u pravokutni spremnik. Razvijene su metode koje daju optimalno iskorištenje spremnika s više od 90%.

Manje istraženo područje je područje pakiranja konkavnih poligona u pravokutni spremnik, kao i pakiranje konkavnih poligona u konkavni spremnik. Spomenuti problem naročito bi bio primjenjiv u odjevnoj i obućarskoj industriji gdje konkavne spremnike mogu predstavljati materijali poput kože koji mogu imati i rupe oštećenja koja se manifestiraju kao rupe. Pakiranje nepravilnih poligona u konkavne spremnike predstavlja znanstveni izazov, od načina računalnog prikaza spremnika i poligona, metode detekcije preklapanja, te izrade algoritma koji bi mogao napraviti ostvariv razmještaj poligona.

Perspektivu za buduća istraživanja ima i metoda izrade giljotinskog pakiranja za konkavne poligone, kao i izrada algoritama za automatsko računalno uklapanje krojnih slika na materijalu s uzorcima.

Ovaj rad je financirala Hrvatska zaklada za znanost projektom 3011 Primjena matematičkog modeliranja i inteligentnih algoritama pri konstrukciji odjeće.

Literatura:

[1] Zelenika R., A. Grilec Kaurić: Ocjena ekonomskog položaja tekstilne i odjevne industrije u Republici Hrvatskoj, Ekonomska misao i praksa **2** (2011.), 543-566

[2] Butorac G. i sur.: Hrvatska tekstilna industrija u Europskoj uniji - konkurentnost i značenje za ukupno gospodarstvo, Tekstil **63** (2014.) 3–4, 113-125

[3] Pavunc M. i sur.: Tekstil u kontekstu održivog razvoja, Tekstil **63** (2014.) 5–6, 195-203.

[4] Ćulahović B.: Sadašnje stanje i razvojne mogućnosti tekstilnog kompleksa Jugoslavije, Tekstil **35** (1986.) 12, 1073-1085

[5] Dyckhoff H.: A typology of cutting and packing problems, European Journal of Operational Research **44** (1990) 2, 145-159

[6] Egeblad J.: Heuristics for Multidimensional Packing Problems, Københavns Universitet, Faculty of Science, Datalogisk Institut, Department of Computer Science, PhD Thesis, 2008

[7] Wäscher G., et al: An improved typology of cutting and packing problems, European Journal of Operational Research **183** (2007) 3, 1109-1130

[8] Kellerer H., et al: Knapsack Problems, Springer, Berlin 2004.

[9] Bennell J. A., J.F. Oliveira: The geometry of nesting problems: A tutorial, European Journal of Operational Research **184** (2008) 2, 397-415

[10] Oliveira J.F.C., J. Ferreira: Algorithms for nesting problems, Applied Simulated Annealing, René V. V. Vidal (Ed.) **396** (1993), 256-273

[11] Segenreich S.A., L.M.P.F.Braga: Optimal nesting of general plane figures: A Monte Carlo heuristical approach, Computers & Graphics **10** (1986) 3, 229-237

[12] Ramesh Babu A., N. Ramesh Babu: A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms, Computer-Aided Design **33** (2001.) 12, 879-891

[13] Mahadevan A.: Optimization in Computer-aided Pattern Packing (Marking, Envelopes), North Carolina State University, Ph.D. Thesis, 1984.

[14] Sato A.K. et al.: An algorithm for the strip packing problem using collision free region and exact fitting placement, Computer-Aided Design **44** (2012) 8, 766-777

[14] Sato A. K., et al: An algorithm for the strip packing problem using collision free region and exact fitting placement, Computer-Aided Design **44** (2012.) 8, 766–777

[15] Ghosh P. K.: An algebra of polygons through the notion of negative shapes, CVGIP: Image Understanding **54** (1991.) 1, 119–144

[16] Burke E. K., et al: Complete and robust no-fit polygon generation for the irregular stock cutting problem, European Journal of Operational Research **179** (2007.) 1, 27–49

[17] Dean H. T., et al: An improved method for calculating the no-fit polygon, Computers & Operations Research **33** (2006.) 6, 1521–1539

[18] Bennell J. A., Song X.: A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums, Computers & Operations Research **35** (2008.) 1, 267–281

[19] Bennell J. A., et al: The irregular cutting-stock problem - a new procedure for deriving the no-fit polygon, Computers & Operations Research **28** (2001.) 271–287

[20] Chazelle B.: The Bottom-Left Bin-Packing Heuristics: An Efficient Implementation, IEEE Transactions on Computers **c–32** (1983.) 8, 697–707

[21] Jakobs S.: On genetic algorithms for the packing of polygons, European Journal of Operational Research **88** (1996.) 1, 165–181

[22] Dowsland K. A., et al: An algorithm for polygon placement using

a bottom-left strategy, European Journal of Operational Research **141** (2002.) 2, 371–381

[23] Art R. C.: An approach to the two dimensional, irregular cutting stock problem, IBM Cambridge Scientific Center, Technical Report 36. Y08, 1966.

[24] Murata H., et al: VLSI module placement based on rectangle-packing by the sequence-pair, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on **15** (1996.) 12, 1518–1524

[25] Tang X. et al: Fast Evaluation of Sequence Pair in Block Placement by Longest Common Subsequence Computation, Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings, Paris 2000., 106–111.

[26] Downsland K. A.: Some experiments with simulated annealing techniques for packing problems, European Journal of Operational Research **68** (1993.) 3, 389–399

[27] Wong D.F., Tang X.: FASTSP: A Fast Algorithm for Block Placement based on Sequence Pair, Design Automation Conference, 2001. Proceedings of the ASP-DAC 2001. Asia and South Pacific, Yokohama 2001., 521 – 526.

[28] Pisinger D.: Denser Packings Obtained in O (n log log n) Time, INFORMS Journal on Computing **19** (2007.) 3, 395–405

[29] Imamichi T.: Nonlinear programming based algorithms to cutting and packing problems, Kyoto University, Ph.D. Thesis, 2009.

[30] Imamichi T., et al: An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem, Discrete Optimization **6** (2009.) 4, 345–361

[31] Liu H., He Y.: Algorithm for 2D irregular-shaped nesting problem based on the NFP algorithm and lowest-gravity-center principle, Journal of Zhejiang University SCIENCE A **7** (2006.) 4, 570–576

[32] Shalaby M. A.: A Particle Swarm Optimization Algorithm for a 2-D Irregular Strip Packing Problem, American Journal of Operations Research **3** (2013.) 2, 268–278

[33] Burke E., Kendall G.: Applying ant algorithms and the no fit polygon to the nesting problem, Advanced Topics in Artificial Intelligence, Springer, Berlin 1999., 453–464.

[34] Fujita K., et al.: Hybrid approach for optimal nesting using a genetic algorithm and a local minimization algorithm, Proceedings of the 19th annual ASME design automation conference **65** (1993.), 477–484.

[35] Kang K., et al: A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem, Applied Mathematics and Computation **219** (2012.) 3, 1287–1299

[36] Hopper E., Turton **:** A genetic algorithm for a 2D industrial packing problem, Computers & Industrial Engineering **37** (1999.) 1, 375–378

[37] Han W., et al: Construction heuristics for two-dimensional irregular shape bin packing with guillotine constraints, European Journal of Operational Research **230** (2013.) 3, 495–504

SUMMARY

Computer-based methods for solving of packing problems at marker making

D. Domović, T. Rolich

In this paper an overview of algorithms for automated marker making problem has been presented. This problem has been a field of interest in computer science within the scope of packing problem algorithms. Packing problem is a problem in which a set of items needs to be placed within the boundaries of a container without overlapping. The goal is to minimize the waste area between polygons, i.e. to reduce the area of a container. Considering the fact packing problems arise in various industries, in this paper an overview of packing problems and its taxonomy is presented, alongside an overview of overlapping detection methods such as raster method, no-fit polygon, direct trigonometry, D-functions and constraint graphs. Also, some of the existing problem solving approaches have been described.

Key words: packing problem, genetic algorithm, overlap detection, automatic marker making, lay plan

University of Zagreb, Faculty of Textile Technology
Department of Basic, Natural and Technical Sciences
Zagreb, Croatia
e-mail: daniel.domovic@ttf.hr; tomislav.rolich@ttf.hr

Received July 2, 2015

Rechnergestützte Methoden für Verpackungsprobleme bei Schnittbildeinfügung

In diesem Artikel wird ein Überblick über Algorithmen für die Untersuchung des zweidimensionalen Problems der automatischen Schnittbildeinfügung dargestellt. Dieses Problem ist ein Interessengebiet in der Informatik im Rahmen der Algorithmen für Verpackungsprobleme gewesen. Verpackungsproblem ist ein Problem, in dem ein Satz Einzelteile innerhalb der Grenzen eines Behälters gesetzt werden muss, ohne sich zu überschneiden. Das Ziel ist, die Fläche des unbenutzten Bereichs zwischen Polygonen zu vermindern, d.h. die Fläche des Behälters zu verringern. Da das Verpackungsproblem in verschiedenen Industriezweigen vorhanden ist, wird in diesem Artikel ein Überblick über Verpackungsprobleme und derer Taxonomie, neben einem Überblick über Überschneidungsnachweismethoden wie Rasterverfahren, No-fit Polygon, direkte Trigonometrie, D-Funktionen und Beschränkungsdiagramme dargestellt. Auch einige der vorhandenen Algorithmen für die Lösung der Verpackungsprobleme werden beschrieben.