



Univerzitet u Novom Sadu
Fakultet tehničkih nauka



Dokumentacija za projektni zadatak

Studenti: Ivanov Maša, SV54/2021
Gašić Dimitrije, SV31/2021

Predmet: Nelinearno programiranje i evolutivni algoritmi

Broj projektnog zadatka: 6

Tema projektnog zadatka: Genetski algoritam, problem 2D rezanja

Opis problema

Dat je materijal širine a i visine b . Materijal treba da se iseče na manje pravougaonike (određenih dimenzija), tako da ostane što manje škarta prilikom rezanja.

- Formirati optimizacioni problem (jasno definisati kriterijum optimalnosti i ograničenja).
- Implementirati rešavanje problema rezanja koristeći genetski algoritam.
- Napraviti korisnički interfejs, gde će korisnik moći da zada dimenzije početnog materijala, kao i željene dimenzije pravougaonika koje treba iseći.
- Grafički predstaviti dobijene rezultate.

Uvod

Optimizacioni problem je definisan funkcijom (0) i podrazumeva traženje njenog maksimuma. Cilj algoritma za traženje rešenja (rasporeda pravougaonika) koje ima najmanje otpada, tj. ima najveću površinu sume pravougaonika za postavljanje. Funkcija (0) pomenutu sumu deli ukupnom površinom materijala za sečenje u svrhu normalizacije brojeva.

$$(0) \max f = \frac{\sum_{i=1}^n w_i h_i}{W * H}$$

Ograničenja problema su predstavljena sledećim nejednakostima:

$$(1) x_i + w_i \leq W \quad y_i + h_i \leq H$$

$$(2) \max \{x_i - (x_j + w_j), x_j - (x_i + w_i), y_i - (y_j + h_j), y_j - (y_i + h_i)\} \geq 0; i \neq j$$

$$(3) 0 < x_i \leq W \quad 0 < y_i \leq H$$

U ograničenjima i formuli W i H predstavljaju zadatu širinu i visinu materijala za sečenje. Simboli w_i i h_i su širina i visina i -tog manjeg pravougaonika dok su x_i i y_i koordinate pozicije tog pravougaonika (odnosno njegovog gornjeg levog ugla) u odnosu na koordinatni početak u gornjem levom uglu materijala za sečenje.

Ograničenje (1) reguliše da se prilikom raspoređivanja novih manjih pravougaonika ne desi da oni “pređu” granice materijala za sečenje. Ograničenje (2) sprečava da se neka dva mala pravougaonika preklope prilikom postavljanja. Ograničenje (3) je ujedno i prirodno ograničenje koje se koristi za regulisanje unosa dimenzija manjih pravougaonika.

Za traženje rešenja je korišćen binarni genetski algoritam sa ruletskom selekcijom, ukrštanjem dve tačke i funkcijama mutacije kao i propagacije najboljih roditeljskih jedinki. Sam genetski algoritam direktno ne menja rotacije pravougaonika za postavljanje nego se to određuje na nižem nivou, u heuristici za raspoređivanje pravougaonika na materijal.

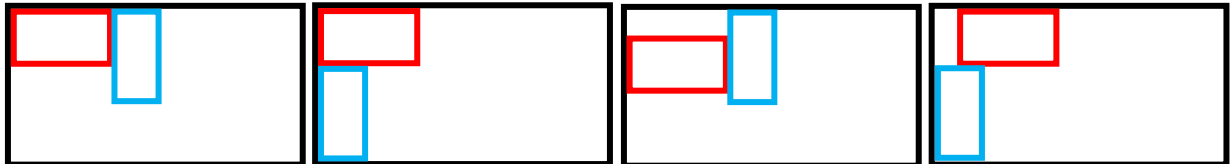
Implementacija

Problem je formulisan tako da unapred znamo koliko će maksimalno manjih pravougaonika biti isečeno iz materijala. Pošto je ovo slučaj, unapred možemo da odredimo redosled pomenutih kvadrata u hromozomu. Ovo je postignuto binarnim brojem kod koga cifre predstavljaju da li će se odgovarajući pravougaonik iseći iz materijala ili ne (1 hoće, 0 neće).

Početnu populaciju hromozoma kreira `generate_initial_chromosomes(population_size=100)` koja koristi `generate_chromosome(N: int)`. Postupak generisanja pojedinačnih hromozoma podrazumeva pravljenje string-a jedinica dužine N i čijih se N/4 random jedinica menja u nule. Za biranje indeksa na kojima će se postaviti nule se koristi `randrange(start, stop)` iz modula `random`.

Jedinka nastaje dekodiranjem hromozoma u niz torki sledećeg formata (x_i, y_i, w_i, h_i) čiji su simboli opisani u uvodu. Proces dekodiranja se vrši u funkciji `(a) decodeChromosome(chromosome)` time što se hromozom čita cifru po cifru i generišu se pravougaonici koji odgovaraju poziciji tih cifara. Ova funkcija vraća dekodiranu jedinku ali i njenu prilagođenost koja se računa formulom (0).

Funkcija koja generiše poziciju pomenutog pravougaonika je `(b) generatePosition(r, table)` gde r predstavlja prethodno definisanu torku gde su x i y jednaki 0. Postoji mogućnost da se prosleđeni pravougaonik rotira ako će to dovesti do bolje iskorišćenosti materijala za sečenje. `(b)` će vratiti izmenjenu torku iz svog argumenta time što će odrediti koordinate x i y, i potencijalno zameniti širinu i visinu (rotiranje). U slučaju da pravougaonik ne može da se iseče iz materijala `(b)` će vratiti None. Sledeći slučajevi ilustruju načine na koje može da se generiše pozicija (plavi se postavlja u odnosu na crveni):

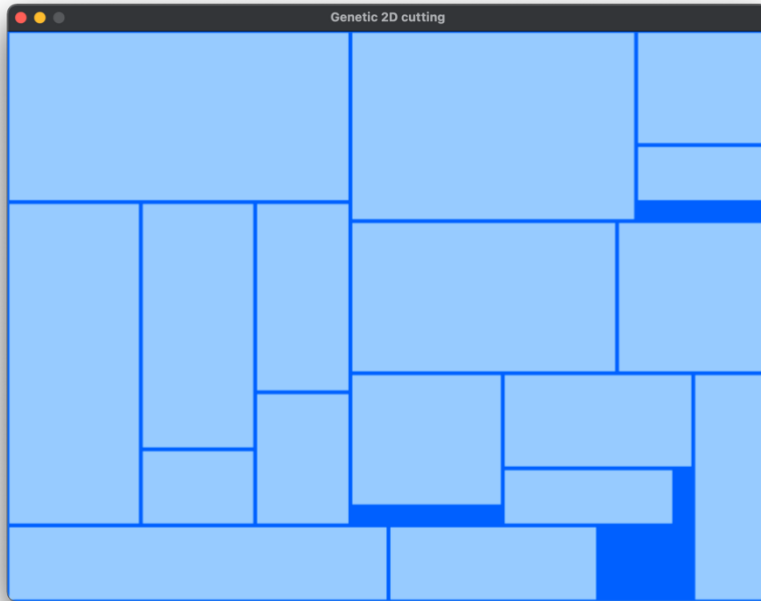


Parove roditelja koji će proizvesti decu se generišu u funkciji `roulette_selection(ranked_parents)` koja prima sortirane roditelje po prilagođenosti. Deca se od parova dobijaju u funkciji `two_point_crossover(pairs)` ukrštanjem dve tačke. Mutacija populacije dece se vrši menjanjem po jednog nasumičnog bita u hromozomima u funkciji `mutation(chromosomes, rate)` i koristi koeficijent od 0.7. Za čuvanje najboljih roditeljskih jedinki se koristi elitizam sa koeficijentom od 0.1 u funkciji `elitism(parents, children, rate, population_size)`.

Kriterijumi za prekid algoritma su: maksimalan broj izvršenih iteracija (100), raspoređeni svi pravougaonici (nebitan procent otpada) i ako najbolja jedinka ima prilagođenost veću od 0.96 uzastopno 10 puta.

Hromozom dobijen genetskim algoritmom se poslednji put dekodira u funkciji `(a)` i prikazuje pomoću `draw_table(table, surface)` koja koristeći modul `draw` iz modula `pygame` prikazuje materijal za sečenje na prethodno kreiranom prozoru.

Performanse



- dimenzije materijala: 400 x 300

pravougaonici za sečenje:	
200 x 40	70 x 60
180 x 90	70 x 50
150 x 100	60 x 130
140 x 80	60 x 40
120 x 60	50 x 100
110 x 40	40 x 120
100 x 50	30 x 90
80 x 80	30 x 70
80 x 70	30 x 60
70 x 170	/

izvršeno iteracija:	17/100 (konvergencija)
vreme izvršavanja:	0.70778 sekundi
procent otpada:	3.17 %
iskorišćeno:	17/19 pravougaonika

Zaključak

Problemi koji spadaju u probleme grafičke optimizacije uvode još nekoliko prepreka. Jedna od njih je definisanje hromozoma jedinice u linearnoj strukturi, premda je ona dosta kompleksnijeg oblika (skup i pozicije oblika u prostoru itd).

Budući da korisnik unosi broj svakog od pojedinačnih pravougaonika, problem ne podrazumeva da sam algoritam generiše dodatne pravougaonike. Naravno, zbog toga se često dešava da je otpad znatno većeg procenta, ali zbog definicije problema to je i dalje najbolje rešenje koje postoji.

Još jedna nuspojava ovog pristupa je mogućnost da krajnje rešenje ne sadrži sve pravougaonike jer ih je nemoguće rasporediti na zadatim dimenzijama materijala. Ako bi bilo potrebno, ovo je moguće sprečiti prekidom programa zbog greške koja bi nastala ako neki pravougaonik ne može da se rasporedi. Ipak, odlučili smo se za pristup da program proizvede najbolje moguće rešenje korisniku, bez ispisa takvih grešaka.