

Projektiranje fasadnih panela pomoću generatora pseudoslučajnih brojeva

Boris Uremović, Mandi Orlić Bachler, Marija Čačić Ptičar

Sažetak

U članku se opisuje generiranje pseudoslučajnih brojeva i primjena generatora pseudoslučajnih brojeva na izradu fasadnih panela. U prvom dijelu rada opisani su Linearni kongruentni generator, Lagged Fibonacci generator i Mersenne Twister generator, dok se u drugom dijelu radu na nekoliko primjera prikazuje primjena generatora pseudoslučajnih brojeva u konstruiranju unikatnih geometrijskih uzoraka fasadnih panela.

Ključni pojmovi: slučajni brojevi, pseudoslučajni brojevi, generatori pseudoslučajnih brojeva, fasadni paneli.

1. Uvod

Tijekom povijesti, prije izuma računala i raznih mehaničkih uređaja, pa i prije nego li su se brojevi označavali simbolima, ljudi su imali potrebu za stvaranjem nasumičnih vrijednosti. Slučajnost pri donošenju odluke ili slučajnost u igrama ljudi su stvarali bacanjem kockice ili novčića. Prije pojave računala, pred stotinjak godina, koristile su se tablice takozvanih brojeva slučajnog uzorka. Međutim, vrlo brzo su se uočili problemi s mnoštvom takvih tablica, posebno jer su se neke znamenke pojavljivale prečesto, a druge pak nedovoljno često. Razvojem računala, počele su se razvijati i razne metode, temeljene na matematičkim formulama i algoritmima, za generiranje slučajnih brojeva. Takvi algoritmi danas

imaju široku primjenu. Primjenjuju se kod simulacija, uzimanje uzoraka, numeričke analize, programiranja, odlučivanja, estetike, kockanja, građevine, kriptografije, pravosudnog sustava,...

U ovom radu bavit ćemo se primjenom pseudoslučajnih brojeva u području graditeljstva.

Projektiranje građevina složen je proces koji zahtjeva koordiniranu suradnju velikog broja sudionika (arhitekti, inženjeri građevinarstva, inženjeri elektrotehnike, inženjeri strojarstva). Proces projektiranja prolazi kroz fazu izrade idejnog rješenja, potom fazu izrade glavnog projekta i na kraju fazu izrade izvedbenog projekta koji prethodi početku gradnje građevine. U današnje vrijeme sve navedene faze projektiranja barem donekle iziskuju upotrebu računala čak i za manje odnosno jednostavne građevine, dok je kod projektiranja složenih građevina upotreba računala neizostavan dio procesa bez kojeg bi cijeli postupak bio otežan, usporan, a u nekim situacijama i gotovo nemoguć. Projektiranje građevina pomoću računala omogućava projektantima značajno veće mogućnosti kreativnog izražavanja, a pogotovo u dijelu projektiranja jedinstvenih građevina koje se svojim oblikom i vanjskim izgledom pokušavaju istaknuti i razlikovati od svoje okoline. S obzirom da se zbog želje za skraćivanjem procesa izgradnje i smanjivanjem troškova pribjegavalo korištenju tipiziranih proizvoda (opeka; crijep i druge vrste pokrovnih elemenata od metala ili betona; tipski montažni elementi; pa čak i betonske konstrukcije koje se iako nemaju unaprijed definirani tipski oblik u konačnici izvede pomoću oplata tipskog oblika), postizanje različitosti među građevinama je dominantno ovisilo o načinu na koji je projektant koristio takve tipizirane proizvode. Tipizirani proizvodi su zbog svojeg tipskog oblika i dimenzija omogućavali proizvođačima da relativno jednostavno, brzo i jeftino proizvedu velike količine proizvoda, dok su izmjene tih proizvoda bile vrlo rijetke, složene, spore i skupe. Razvitkom modernih tehnologija proizvodnje pojavljuju se nove mogućnosti izrade građevinskih elemenata, te postaje moguće proizvoditi građevinske elemente koji posjeduju veću dozu međusobne različitosti iako zadržavaju određene zajedničke karakteristike. Primjer toga su i montažni fasadni paneli kojima se često pokrivaju fasade tradicionalnog oblika i karakteristika (slika 1). Na slici 1. je u prednjem planu moguće vidjeti primjer zgrade napravljene od klasičnih tipskih armiranobetonskih montažnih elemenata (stupovi, grede, nosači,...) čija je fasada pokrivena alumi-nijskim fasadnim elementima s Voronoj uzorkom. Iako ne postoje dva panela s istim rasporedom otvora, njihova izrada je bila relativno jednostavna i brza zbog upotrebe računalne tehnologije izrade panela koja je umjesto ručnog rezanja i bušenja pojedinog panela koristila rezanje plazmom ili laserom, a sve temeljem računalno izrađenog uzorka dobivenog korištenjem generatora pseudoslučajnih brojeva.



Slika 1. Trgovački centar Galeria Warmińska u Poljskoj

2. Slučajni i pseudoslučajni brojevi

Na početku se prisjetimo definicije niza.

Definicija 1. *Neka je S neki skup brojeva. Funkciju $a: \mathbb{N} \rightarrow S$ zovemo nizom brojeva iz S . Elementi a_1, a_2, \dots, a_n zovu se članovi niza, a član a_n opći član niza.*

Slučajan broj, kao što i naziv sugerira, je broj koji je slučajno izabran iz skupa brojeva. Npr. broj izvučen na lotu je slučajan broj. Za takav broj možemo reći da je nepredvidljiv zato što ga ne možemo predvidjeti na temelju prethodno izvučenih brojeva, ali i da je uniforman, jer je vjerojatnost izvlačenja svakog broja iz bubnja jednaka.

Razlikujemo dvije vrste slučajnih brojeva „prave” slučajne brojeve i pseudoslučajne brojeve.

„Pravi” slučajni brojevi su brojevi koji zadovoljavaju uvjete nepredvidljivosti i uniformnosti. Za niz brojeva kažemo da je nepredvidljiv ako na temelju poznatih članova a_1, a_2, \dots, a_n niza ne možemo predvidjeti sljedeći član a_{n+1} , bez obzira koliko velik bio n . Za niz brojeva kažemo da je uniforman ako je učestalost pojavljivanja različitih vrijednosti članova niza približno ista.

Pseudoslučajni brojevi su brojevi koji su generirani uz pomoć računala, odnosno algoritma. U tu svrhu računalo koristi početnu vrijednost

(engl. *seed value*¹), koja je „pravi” slučajni broj, te pomoću rekurzivnih aritmetičkih i/ili logičkih formula dobivamo pseudoslučajni niz brojeva. Iako takvi nizovi izgledaju kao slučajni, ipak ih ne možemo smatrati u potpunosti slučajnima, jer su generirani determinističkim procesima, koje po potrebi možemo ponoviti. Ovakvi nizovi su periodički, odnosno nakon nekog vremena članovi niza će se ponoviti.

2.1. Generiranje slučajnih brojeva

Generiranje slučajnih brojeva podrazumjeva generiranje niza brojeva koji nemogu unaprijed biti predviđeni. Metode generiranja slučajnih brojeva općenito se mogu podijeliti u dvije kategorije: generatori „pravih” slučajnih brojeva i generatori pseudoslučajnih brojeva.

Generatori „pravih” slučajnih brojeva (eng. *True Random Number Generators*, TRNG) generiraju brojeve na temelju fizičkih (prirodnih) procesa. Kod takvih generatora slučajni broj u nizu neovisan je o svom prethodniku. Prednost ovih generatora je što su nedeterministički i nisu periodički, ali zato izrazito sporo generiraju slučajan niz brojeva.

Generatori pseudoslučajnih brojeva (eng. *Pseudo-Random Number Generators*, PRNG) generiraju niz (naizgled) slučajnih brojeva koristeći algoritme. Ovako generirani pseudoslučajni nizovi, kao i njihovi podnizovi, trebali bi biti ravnomjerno raspoređeni u prostoru, podnizovi ne bi smijeli biti u korelaciji i period generatora mora biti što je moguće veći. Duljina niza prije nego što se niz počne ponavljati zove se period (najčešće se kreće od 2^{31} do 2^{64}) [1]. Mane ovih generatora su determinističnost i periodičnost, ali je zato njihova velika prednost brzina generiranja slučajnih brojeva.

2.1.1 Linearni kongruentni generator

Prisjetimo se sljedećih definicija.

Definicija 2. *Prirodni broj $p > 1$ je prost broj ako ima točno dva prirodna djelitelja, broj 1 i samog sebe.*

Definicija 3. *Ako cijeli broj $m \neq 0$ dijeli razliku $a - b$, onda kažemo da je a kongruentno b modulo m i pišemo $a \equiv b \pmod{m}$. U protivnom, kažemo da a nije kongruentno b modulo m .*

Linearni kongruentni generator (LKG) jedan je od najstarijih i najpoznatijih generatora pseudoslučajnih brojeva. Generator je unaprijed određen početnom vrijednosti (sjemenom) x_0 , multiplikatorom $a \in \mathbb{N}$,

¹Ovisno o literaturi za ovu vrijednost koriste se još i nazivi sjeme i ključ.

inkrementom $c \in \mathbb{N}_0$ i operatorom cjelobrojnog ostatka dijeljenja modlom $m \in \mathbb{N}$, koji su zadani od strane korisnika. Neka je $a < m$, $c < m$, $x_0 \in \mathbb{N}_0$, $x_0 < m$ početna vrijednost, tada se svaki sljedeći član dobiva iz prethodnog pomoću izraza

$$x_{n+1} \equiv a \cdot x_n + c \pmod{m}, \quad (1)$$

pri čemu je $0 \leq x_n \leq m - 1$. Dobiveni niz naziva se Lehmerov niz². Ovako dobiveni niz je periodičan, a članovi niza su brojevi od 0 do $m - 1$, što znači da će se najkasnije nakon prvih m članova niza članovi početi ponavljati. Međutim, pripazi li se na odabir brojeva m , a i c period niza može biti dovoljno velik, pa će se ponavljane vrijednosti u nizu dogoditi rijetko.

Algoritam generatora pseudoslučajnih brojeva možemo implementirati kao program u bilo kojem programskom jeziku. Rezultat tog programa bit će niz pseudoslučajnih brojeva. Pogledajmo na sljedećem primjeru kako u programu Maxima možemo ispisati 10 članova pseudoslučajnog niza dobivenog izrazom (1).

Primjer 1. *Uzmimo da su $x_0 = 12$, $a = 2$, $c = 4$ i $m = 15$. Tada generator (1) možemo u Maximi zapisati na sljedeći način:*

```
(%i3) x_0 : 12$
      a : 2$
      c : 4$
      m : 15$
      lkg() := x_0 : mod(a * x_0 + c, m)$;
      makelist(lkg(), 10);
(%o3) [13, 0, 4, 12, 13, 0, 4, 12, 13, 0]
```

Period dobivenog niza je 4, jer nakon četvrtog broja u nizu vrijednosti niza počinju se ponavljati.

Odaberemo li ulazne vrijednosti u generatoru tako da je $a - 1$ višekratnik svakog prostog faktora modula m , da je $a - 1$ djeljiv s 4 ako je m djeljiv s 4 i da je vrijednost multiplikatora jednaka 2^w , pri čemu je w duljina riječi u računalu ili najveći prosti broj prikaziv u računalu, tada ćemo dobiti niz najduljeg perioda čije se vrijednosti generiraju pomoću izraza:

$$x_{n+1} \equiv (2^k + 1) \cdot x_n + c \pmod{2^w}, \quad (2)$$

pri čemu je $2 \leq k \leq w$, $k \in \mathbb{N} [2, 3]$.

²Linearni kongruentni generator 1948. godine kreirao je američki matematičar Derrick Henry Lehmer.

U literaturi [4]-[6] može se pronaći velik broj primjera dobrih generatora. Ovdje izdvajamo jedan primjer gdje se za odgovarajuće ulazne konstante dobiva niz čiji je period $2^{31} - 2 = 2147483646$ [6].

Primjer 2. *Ulazni parametri su: $x_0 = 1$, $a = 7^5 = 16807$, $c = 0$ i $m = 2^{31} - 1$.*

```
(%i3)  x_0 : 1$
        a : 7^5$
        c : 0$
        m : 2^31 - 1$
        lkg() := x_0 : mod(a * x_0 + c, m)$;
        makelist(lkg(), 10);
(%o3)  [16807, 282475249, 1622650073, 984943658, 1144108930,
        470211272, 101027544, 1457850878, 1458777923, 2007237709]
```

Općenito, linearni kongruentni generator možemo podijeliti na tri slučaja:

1. Generator kod kojeg je $m = 2^k$, $c \in \mathbb{Z}$ i $c > 0$. Ovakav generator postiže puni period 2^k ako je konstanta $a = 1 \pmod{4}$ i c je neparan broj. Prednost ovog generatora je što je izrazito brz.
2. Generator kod kojeg je $m = 2^k$, $c = 0$ zove se multiplikativni kongruentni generator. Najveći period postiže se za 2^{k-2} i to ako je $a = 3 \pmod{8}$ ili $a = 5 \pmod{8}$.
3. Generator kod kojeg je $m = p$ prost broj. Puni period iznosi $p - 1$ i postiže se ako je a prost broj. Ovakav generator prikazan je u primjeru 2.

LKG generator koristio se u mnogim softverima, pogotovo multiplikativni kongruentni generator s parametrima $m = 2^{31} - 1$ i $a = 16807$ koji je bio algoritam ugrađene funkcije `minstd_rand0` u programskom jeziku C++.

2.1.2 Lagged Fibonacci generator

Definicija 4. *Fibonaccijev niz definira se na rekurzivan način formulom*

$$F_{n+1} = F_n + F_{n-1}$$

uz početne vrijednosti $F_1 = 1$, $F_2 = 1$.

Lagged Fibonacci generator (LFG) generira niz slučajnih brojeva koristeći izraz:

$$F_n = F_{n-i} \bullet F_{n-j} \pmod{m},$$

pri čemu su $i, j \in \mathbb{Z}$, $0 < i < j$, a operacija \bullet može biti bilo koja binarna operacija $(+, -, \times, \pmod{m}, \text{XOR}, \dots)$. Najčešće se za operaciju \bullet uzima zbrajanje modulo m , pri čemu je $m = 2^k$. Tada je generator oblika:

$$F_n = F_{n-i} + F_{n-j} \pmod{2^k}.$$

Početni uvjet generatora je niz brojeva X_0, \dots, X_{i-1} koji se mogu generirati npr. linearnim kongruentnim generatorom. Primjeri dobrih LFG generatora su:

1. $i = 17, j = 5, k = 31$, period je $P \approx 2^{47}$,
2. $i = 55, j = 24, k = 31$, period je $P \approx 2^{85}$.

U usporedbi s LKG generatorom LFG generator pri generiranju niza ima bolja slučajna svojstva (pogotovo perioda generiranih nizova). Međutim, njegov je nedostatak što zahtjeva veći memorijski prostor [1].

2.1.3 Mersenne Twister generator

Definicija 5. *Mersennov prost broj je prost broj oblika $M_n = 2^n - 1$.*

Najkorišteniji generator pseudoslučajnih brojeva je Mersenne Twister generator kojeg su kreirali Makoto Matsumoto i Takuji Nishimura 1998. godine. Algoritam je implementiran kod velikog broja softvera kao što su Matlab, Microsoft Excel, Python, R, Wolfram Mathematica, C++, Maxima, ... Najčešće se koristi Mersenne Twister algoritam (MT19937) kod kojeg je dužina perioda jednaka Mersennovom prostom broju $2^{19937} - 1$ te koristi 32-bitne računalne riječi.

U nastavku rada kroz nekoliko primjera pokazat ćemo kako se pomoću generatora pseudoslučajnih brojeva mogu konstruirati unikatni geometrijski uzorci fasada.

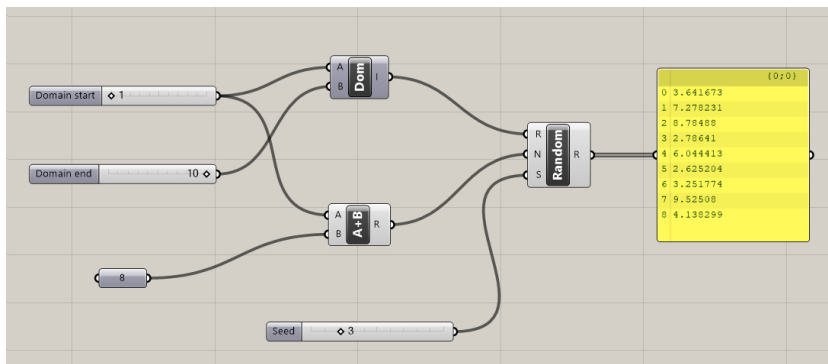
3. Projektiranje fasadnih panela

Za primjer u ovom radu je uzeta pravokutna građevina jednostavne konstrukcije koja se sastoji samo od nosivih elemenata stropnih ploča i stupova. Takav tip konstrukcije je vrlo čest kod garaža ili poslovnih objekata s obzirom na otvorenost tlocrta i manjak zidova, pa je takve objekte jednostavnije prilagoditi različitim potrebama. Na navedenoj građevini je predviđena ovojnica zgrade izrađena od staklenih panela koji pokrivaju fasadu od vrha do dna po cijelom njenom obodu. Takve zgrade

s ostakljenim fasadama vrlo često imaju problem s velikom količinom svjetla koja ulazi u objekt i to prvenstveno na južnoj, ali i na istočnoj i zapadnoj strani. Tradicionalno, takve ostakljene fasade se (na sjevernoj Zemljinoj polutci) s južne strane zasjenjuju sjenilima koja slabije propuštaju svjetlost, dok fasade na istočnoj i zapadnoj strani zasjenjujemo sjenilima koja imaju veće otvore i propuštaju više svjetlosti od onih na južnoj strani građevine. Na sjevernoj fasadi nije uobičajeno postavljanje sjenila jer vrlo često postoji nedostatak prirodnog svjetla, no može se pojaviti potreba za sjenilom ukoliko sa sjeverne strane postoji drugi objekt od kojeg se odbija svjetlost (najčešće objekt izrazito svijetle fasade ili s ostakljenom fasadom). Na građevini iz primjera sjenila će biti postavljena na sve četiri fasade, fasade će se sastojati od pravokutnih panela koji će biti izrađeni od aluminijskog ili čeličnog lima. Širine panela na pojedinoj fasadi će biti određene na temelju veličine zgrade odnosno širine i dubine zgrade, te broja panela po širini odnosno dubini fasade, a važno još je naglasiti da je zbog estetskih i tehničkih razloga visina svih panela na svim fasadama jednaka. Otvori na panelima će biti izrađeni nekom od tehnologija za strojno rezanje metala kao što su lasersko rezanje ili rezanje plazmom, i na sve četiri fasade paneli će imati drugačiji oblik otvora temeljen na geometrijskim oblicima čiji će izgled ili položaj biti djelomično određen korištenjem generatora pseudoslučajnih brojeva.

Sva sjenila su napravljena korištenjem programskog alata Rhinoceros 3D i dodatka za vizualno programiranje Grasshopper.

Za programiranje pomoću VPL alata Grasshopper korisnik ne mora posjedovati znanje programiranja korištenjem pisanog odnosno tekstualnog koda koji koriste programski jezici kao što su C++, Python ili neki drugi od često korištenih programskih jezika. Za korištenje Grasshopper VPL-a nužno je posjedovati sposobnost logičkog rasuđivanja, te naravno znanje algebre i geometrije, a uz navedeno potrebno je poznavati dijelove VPL Grasshopper programskog koda. Programski kod u Grasshopper VPL alatu se sastoji od grafičkih elemenata čvorova i veza. Čvorovi mogu predstavljati (1) ulazne parametre ili (2) naredbe odnosno operacije koje obrađuju ulazne parametre, a čiji rezultati mogu biti brojevi ili geometrijski. Čvorovi su najčešće pravokutnog oblika, a ukoliko čvor predstavlja operaciju onda se s lijeve strane pravokutnika nalaze točke ulaza podataka, a s desne strane pravokutnika se nalaze točke izlaza podataka. Kod čvorova koji predstavljaju ulazne parametre nema točaka za ulaz nego postoje samo točke za izlaz s lijeve strane. Veze služe za povezivanje čvorova i prijenos podataka između njih, pa sukladno tome mogu povezivati isključivo točke izlaza podataka na jednom čvoru s točkama ulaza podataka na drugom čvoru, a nipošto ne dvije ulazne točke ili dvije izlazne točke. Na slici 2 prikazan je primjer jednostavnog algoritma za generiranje pseudoslučajnog niza brojeva. U konkretnom primjeru opera-



Slika 2. *Primjer jednostavnog algoritma za generiranje pseudoslučajnog niza brojeva koji je izrađen pomoću VPL alata Grasshopper*

cija *Random* generira niz pseudoslučajnih brojeva iz raspona $[1, 8]$ (operacija *Dom* - *Domain*), koji ima 9 članova, a što je promjenjivi parametar dobiven zbrojem (operacija *A+B*) donje granice raspona (prirodni broj iz raspona $[1, 10]$ prikazan u obliku pomičnog klizača - u ovom primjeru je odabran broj 1) i stalnog parametra (prirodni broj 8)). Sjeme raspona (*S* - *Seed*) je određeno promjenjivim parametrom (prirodni broj iz raspona $[1, 11]$ prikazan u obliku pomičnog klizača - u ovom primjeru je odabran broj 3). U žutom čvoru (okvir za prikaz podataka) prikazan je generirani niz pseudoslučajnih brojeva sukladan ulaznim parametrima i postavljanim operacijama i vezama.

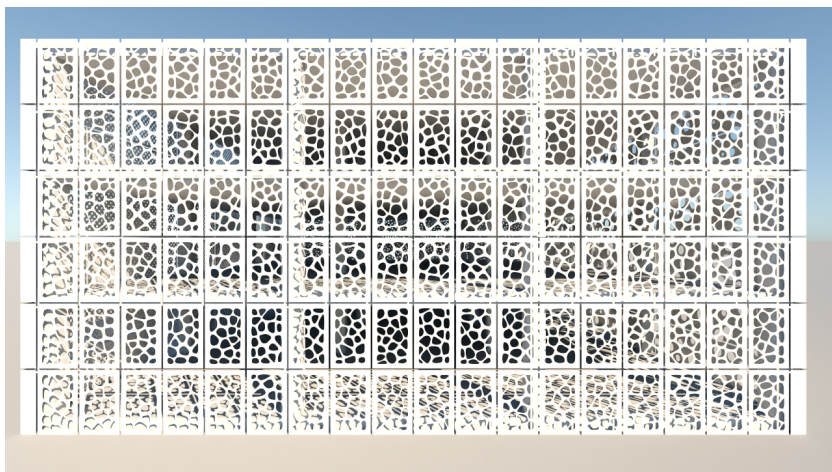
Neka od sjenila su napravljena kao zasebni paneli kod kojih algoritam panela nije međusobno povezan, neka sjenila su napravljena na način da je algoritam primijenjen na cijelu fasadu građevine koja je onda podijeljena na zasebne panele, dok je kod nekih sjenila algoritam formiran na način da su paneli formirani neovisno jedan od drugog, no onda su pojedini elementi svih panela mijenjani u odnosu na točke koje pripadaju cijeloj fasadi. S obzirom da sjenila moraju zadovoljiti i neke tehničke uvjete kao što su primjerice omogućavanje sigurnog pričvršćivanja za fasadu objekta ili osiguravanje dostatne konstruktivne čvrstoće koja sprječava deformacije panela pod utjecajem vanjskih faktora poput vjetera, unutar algoritma definirani su slijedeći parametri: (1) širina, dubina i visina zgrade; (2) visina fasade - fasada ne mora biti iste visine kao i zgrada (3) broj panela po širini objekta; (4) broj panela po dubini objekta; (5) broj panela po visini fasade; (6) razmak između panela. Sa svrhom ispunjavanja tehničkih uvjeta, uz navedene parametre je na nekim od panela definiran i zaštitni pojas (okvir) unutar kojeg neće biti

otvora. Svi navedeni parametri određuju efektivno područje pojedinog panela, odnosno područje na panelu na kojem će biti izrezani otvori. Za izradu modela zgrade, i svih fasadnih panela bit će korišten jedan sveobuhvatni algoritam, no formiran na takav način da je moguće korigirati panele jedne fasade neovisno o panelima drugih fasada. Jedini parametar koji je jednak kod svih fasada je visina panela, te je na taj način osigurano da vodoravne reške na svim fasadama budu na istoj visini i da u kontinuitetu okružuju cijelu građevinu. Visina svih panela je izračunata na temelju parametra „visina fasade” i parametra „broj elemenata po visini fasade”, te se navedeni podatak koristi u svakoj od četiri fasade objekta za određivanje veličine pojedinog panela.

Prva fasada sastoji se od 108 panela raspoređenih u 6 redova po 18 panela. Geometrijska osnova svih otvora na panelima je Voronj dijagram³ temeljen na nasumično postavljenim točkama po efektivnoj površini panela. Svaki panel je oblikovan u slijedećim koracima: (1) početni parametri „dubina građevine” i „broj panela po dubini građevine na prvoj fasadi” zajedno sa zajedničkom visinom panela određuju veličinu pravokutnog panela; (2) dodavanjem zaštitnog pojasa pomoću promjenjivog parametra (prirodni broj iz intervala [5, 20]) određena je efektivna površina na kojoj će biti izrađeni otvori; (3) pomoću generatora pseudoslučajnih brojeva na efektivnu površinu panela postavljeno je pseudoslučajno polje točaka, a broj točaka je određen promjenjivim parametrom (prirodni broj iz raspona [10, 30]); (4) na temelju postavljenih točaka i ruba efektivne površine panela izrađen je Voronj dijagram; (5) svaka od ćelija Voronj dijagrama uniformno je umanjena za isti faktor određen promjenjivim parametrom (racionalni broj iz raspona [0.1, 0.9]), a kao bazne točke smanjenja korištene su točke težišta svake pojedine ćelije; (6) na temelju ruba svake smanjene ćelije (poligon) kreirana je NURBS krivulja⁴ kojoj su kontrolne točke bile vrhovi poligona ćelije i koje su u konačnici predstavljale otvore na panelu. Na ovaj način definiran je izgled jednog panela, a izgled ostalih panela dobiven je izmjenom početnih vrijednosti (sjemena) za generator pseudoslučajnih brojeva iz koraka 3. Raznolikost početnih vrijednosti je postignuta generiranjem niza prirodnih brojeva kojem je početni broj niza određen kao parametar jer u protivnom ne bi postojala mogućnost izbora konačnog oblika. Na slici 3. je prikazan izgled svih panela prve fasade. Druga fasada sastoji se od 18 panela raspoređenih u 6 redova po 3 panela. Geometrijska osnova svih otvora na panelima je polje mnogokuta različitih veličina i kutova rotacije, temeljeno na nasumično postavljenim točkama po efektivnoj površini panela. Svi mnogokuti na drugoj fasadi imaju jednak broj stranica i jednak početni polumjer opisane kružnice. Svaki

³O Voronj dijagramu pogledati [7]

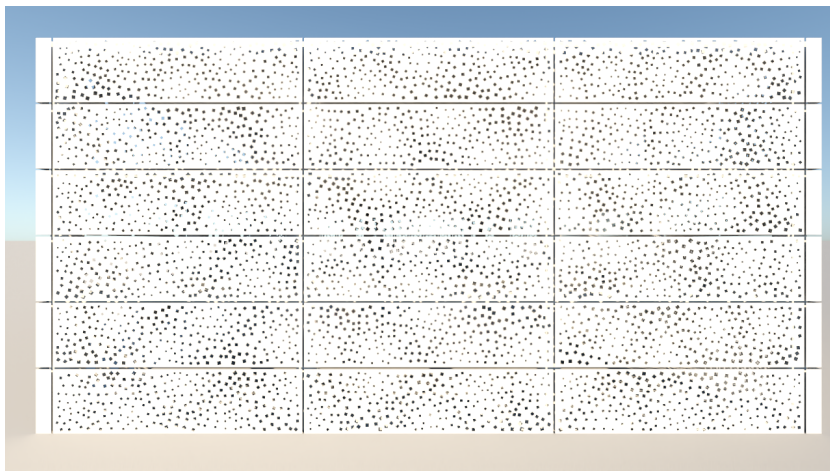
⁴O NURBS krivulji pogledati [8].



Slika 3. *Izgled panela prve fasade*

panel je oblikovan u sljedećim koracima: (1) početni parametri „dubina građevine” i „broj panela po dubini građevine na prvoj fasadi” zajedno sa zajedničkom visinom panela određuju veličinu pravokutnog panela; (2) kao i kod prve fasade, dodavanjem zaštitnog pojasa pomoću promjenjivog parametra (prirodni broj iz intervala $[5, 20]$) određena je efektivna površina na kojoj će biti izrađeni otvori; (3) pomoću generatora pseudoslučajnih brojeva na efektivnu površinu panela postavljeno je prvo pseudoslučajno polje točaka, a broj točaka je određen promjenjivim parametrom (prirodni broj iz raspona $[10, 500]$). To prvo polje točaka će biti korišteno za postavljanje mnogokuta kojima će te točke poslužiti kao središta za postavljanje, rotaciju i smanjivanje; (4) pomoću generatora pseudoslučajnih brojeva na efektivnu površinu panela postavljeno je drugo pseudoslučajno polje točaka, a broj točaka je određen promjenjivim parametrom (prirodni broj iz raspona $[1, 10]$). To drugo polje točaka će biti korišteno za određivanje faktora umanjenja pojedinog mnogokuta u ovisnosti o udaljenosti središta mnogokuta od točaka drugog polja; (5) na temelju točaka prvog polja točaka nacrtani su mnogokuti kojima je ulaznim parametrima određen jednak početni polumjer (prirodni broj iz raspona $[5, 10]$ i broj stranica (prirodni broj iz raspona $[3, 12]$); (6) svi mnogokuti su rotirani oko svojeg središta, a kut rotacije u stupnjevima je određen za svaki mnogokut pomoću PRNG-a (racionalni broj iz raspona $[0, 90]$); (7) svi mnogokuti su umanjeni oko svojeg središta za faktor određen na temelju udaljenosti točaka prvog polja točaka (središta mnogokuta) i drugog polja točaka. Na slici 4.

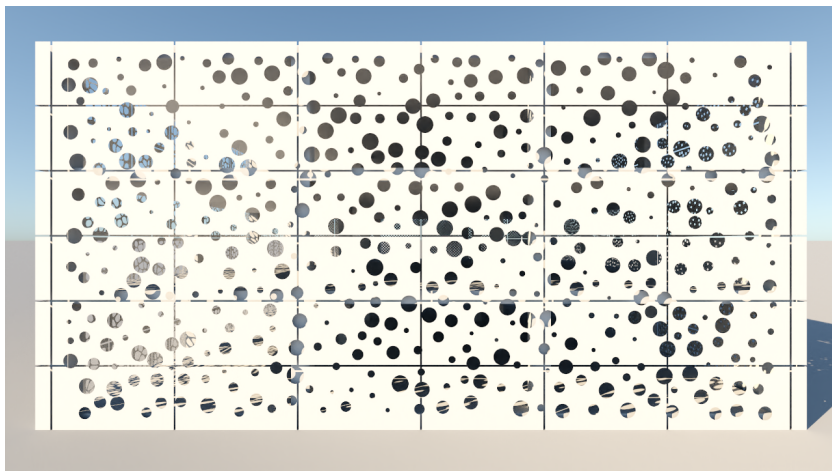
je prikazan izgled svih panela druge fasade. Treća fasada sastoji se od



Slika 4. Izgled panela druge fasade

36 panela raspoređenih u 6 redova po 6 panela. Geometrijska osnova svih otvora na panelima je polje kružnica različitih veličina koje su pomoću pseudoslučajnog polja točaka raspoređene po cijeloj fasadi, a potom razdijeljene na panele. Za razliku od prethodne dvije fasade, raspored kružnica nije ovisan o broju panela već je ovisan o veličini fasade i veličini zaštitnog pojasa. Prilikom određivanja ulaznih parametara cilj je bio pronaći parametre kod kojih ne dolazi do presijecanja kružnica, jer bi presijecanjem kružnica prilikom rezanja moglo doći do izrezivanja većih površina panela odnosno čitavih dijelova panela koji bi se nalazili između tih spojenih odnosno presječenih kružnica. Paneli su oblikovani u sljedećim koracima: (1) pomoću parametara „širina zgrade i „visina fasade” određena je veličina fasade koja je pokrivena panelima; (2) dodavanjem zaštitnog pojasa pomoću promjenjivog parametra (prirodni broj iz raspona [10,30]) određena je efektivna površina na kojoj će biti izrađeni otvori; (3) pomoću PRNG-a je na efektivnu površinu fasade postavljeno polje točaka koje će poslužiti kao središta kružnica (4) polumjeri kružnica određeni su pomoću PRNG-a kojem je određen raspon najvećeg i najmanjeg polumjera kružnice (racionalni broj iz raspona [5,20]); (5) pomoću polja točaka i polumjera nacrtane su kružnice (6) pomoću parametara „broj panela po širini građevine” i „broj panela po visini fasade” određena je veličina pravokutnog panela; (7) na temelju kružnica i rasporeda panela njihovim presijecanjem određen je konačni izgled svakog pojedinog panela. Na slici 5. je prikazan izgled svih panela

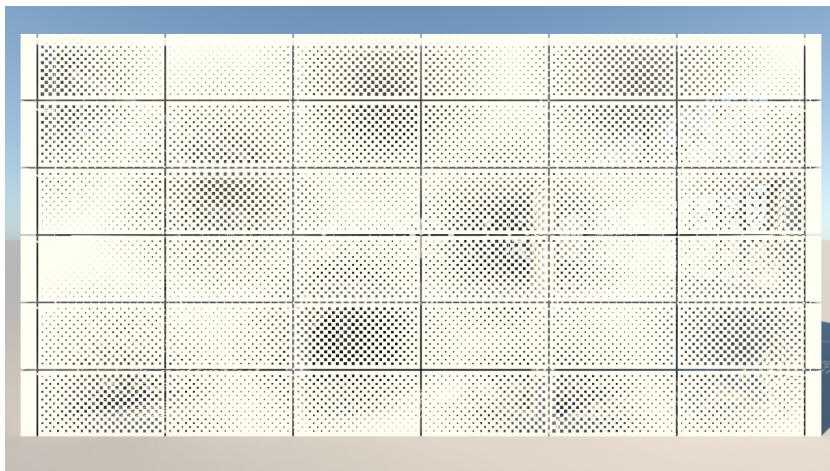
druge fasade. Vidljivo je da su neke kružnice sastavni dio više panela



Slika 5. Izgled panela treće fasade

odnosno da raspored kružnica nije ovisan o broju panela odnosno njihovom rasporedu. Četvrta fasada sastoji se od 36 panela raspoređenih u 6 redova po 6 panela. Geometrijska osnova svih otvora na panelima je pravilno polje kvadrata različitih veličina. Veličina svakog pojedinog kvadrata ovisna je o nasumično postavljenom polju točaka preko cijele fasade, odnosno o udaljenosti svakog pojedinog kvadrata od točaka polja. Svaki panel je oblikovan u sljedećim koracima: (1) početni parametri „širina građevine” i „broj panela po širini građevine na četvrtoj fasadi” zajedno sa zajedničkom visinom panela određuju veličinu jednog pravokutnog panela; (2) kao i kod ostalih fasada dodan je zaštitni pojas na svaki panel pomoću promjenjivog parametra (prirodni broj iz raspona [5, 10] i određena je efektivna površina na kojoj će biti izrađeni otvori; (3) osnova za otvore je pravilno polje kvadrata kojem je uklonjen svaki drugi kvadrat - na taj način je smanjena površina otvora no povećana je krutost panela zbog veće površine panela koji ostaje nakon izrezivanja otvora - polje je određeno promjenjivim parametrom (racionalni broj iz raspona [1, 50]) - taj parametar je morao biti odabran na način da polje kvadrata koje nastane s tom veličinom pojedine ćelije ima neparni broj kvadrata i po visini i po širini panela jer u protivnom nije moguće postići dijagonalni uzorak uklanjanjem svakog drugog kvadrata; (4) nakon definiranja polja kvadrata uklonjen je svaki drugi kvadrat da bi se postigao dijagonalni uzorak na svakom panelu; (5) pomoću PRNG-a je na površini fasade napravljeno pseudoslučajno polje točaka; (6) navedeno

polje točaka je kao i kod treće fasade poslužilo za određivanje faktora umanjjenja kvadrata temeljem udaljenosti središta kvadrata i točaka polja što je dalo konačni izgled panela i cijele fasade. Na slici 6. je prikazan

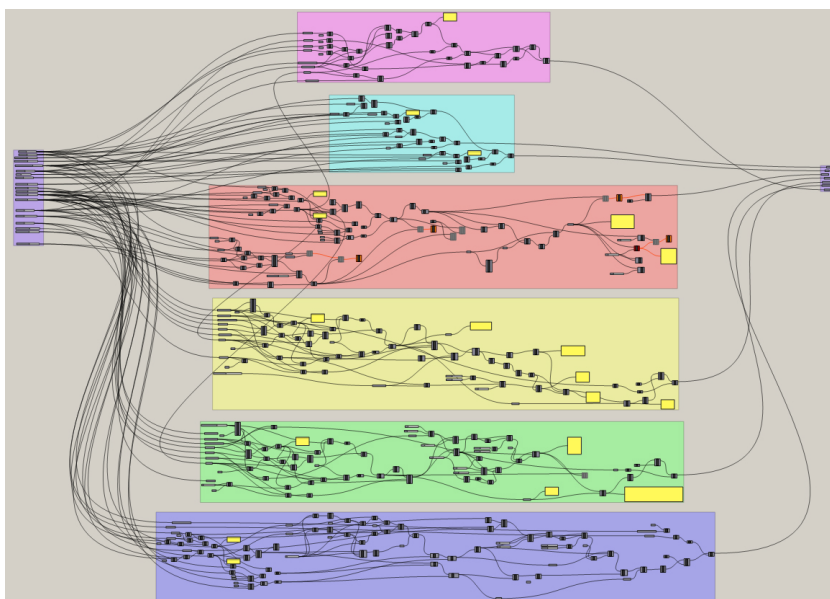


Slika 6. Izgled panela četvrte fasade

izgled svih panela četvrte fasade. Sve četiri fasade iz ovog primjera su izrađene algoritmom koji u nekom aspektu koristi PRNG. U službenoj dokumentaciji proizvođača za Grasshopper nije moguće pronaći točan podatak o tome koja vrsta PRNG-a je implementirana u njegove naredbe koje imaju „nasumičnu” komponentu. Poznat je samo podatak da je programski kod Grasshopper VPL dodatka izrađen korištenjem Microsoft Visual Basic Scripta, pa je pretpostavka da koristi istu vrstu PRNG-a. Nažalost niti u službenoj dokumentaciji za MS VBScript nema egzaktnog podatka koja vrsta PRNG-a je implementirana u njegove RND naredbe. Prema dostupnim informacijama sa službenih i neslužbenih internetskih foruma i za VBScript i za Grasshopper 3D, korisnici su primijetili velike sličnosti u „ponašanju” RND naredbi oba alata, pa djeluje da pretpostavka da oba alata koriste istu vrstu PRNG-a ima čvrste temelje, no za provjeru navedene pretpostavke bilo bi potrebno provesti više različitih provjera kako bi se ta pretpostavka pouzdano mogla potvrditi, a što izlazi iz okvira ovog rada, no predstavlja interesantnu temu za daljnje istraživanje.

Slika 7. prikazuje značajno umanjeni prikaz cjelokupnog algoritma prikazanog na Grasshopper platnu, koji se sastoji od 668 čvorova, od kojih su većina operacije, a manji dio su ulazni parametri. Zbog lakšeg snalaženja pojedini čvorovi su grupirani u različite grupe sukladno dijelu

geometrije modela, pa tako čvorovi iz žutog okvira predstavljaju dio algoritma za izradu druge fasade, čvorovi iz zelenog okvira predstavljaju dio algoritma za izradu prve fasade, čvorovi iz crvenog okvira predstavljaju dio algoritma za izradu treće fasade, a čvorovi iz plavog okvira predstavljaju dio algoritma za izradu četvrte fasade. Čvorovi iz svijetloplavog okvira predstavljaju dio algoritma za izradu nosive konstrukcije zgrade, odnosno stupova i međukatnih konstrukcija, a čvorovi iz ljubičastog okvira predstavljaju dio algoritma za izradu kutnih fasadnih panela koji nemaju uzorak no potrebni su s građevinskog aspekta cjelovitosti fasade. Preostali čvorovi iz manjih okvira s desne i lijeve strane predstavljaju ulazne parametre (okvir s lijeve strane) i izlaznu geometriju (čvorovi s desne strane). Pri izradi navedenog algoritma nisu zasebno programi-



Slika 7. Umanjeni prikaz cjelokupnog algoritma

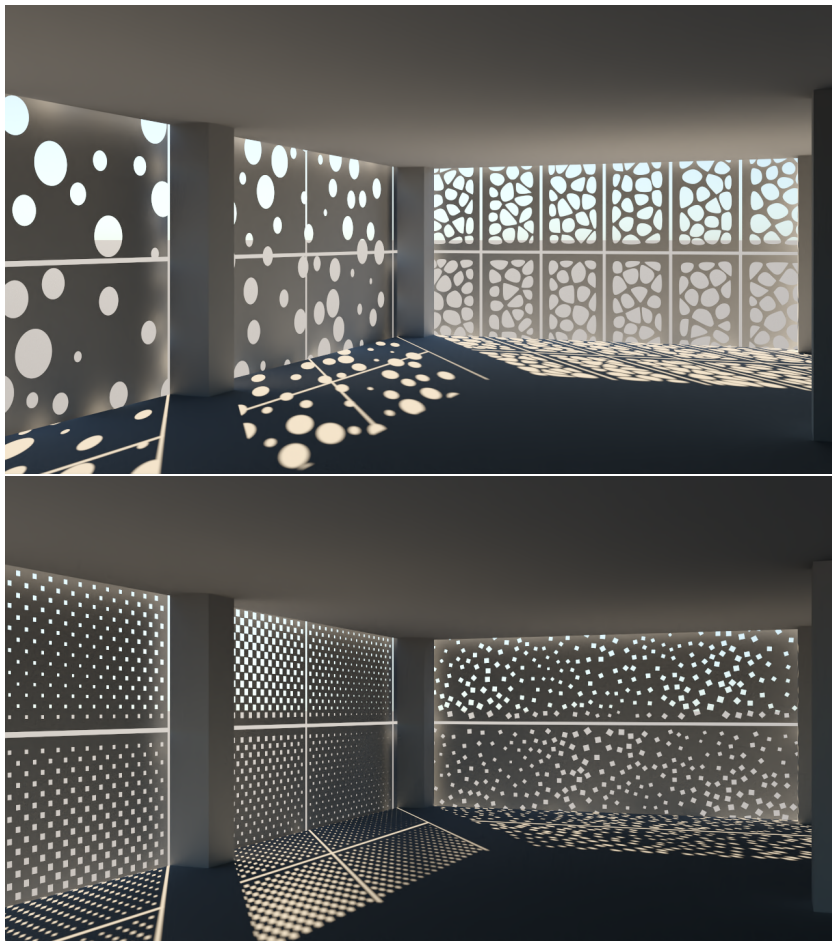
rani odnosno kodirani PRNG-ovi, već su korištene integrirane opracije za generiranje pseudoslučajnog niza brojeva (*Random Number Sequence*), generiranje pseudoslučajnog polja točaka u ravlini (*Populate 2D*), te za pseudoslučajnu izmjenu rasporeda poznatog niza elemenata (*Jitter*).

S obzirom da je zbog velikog broja elemenata na platnu nemoguće čitljivo prikazati cjelokupni algoritam na slici 8. prikazan je isječak algoritma za generiranje kvadrata na panelima druge fasade, a za što je korištena izrada dva pseudoslučajna polja točaka (jednog za središta kva-

Slika 8. Isječak algoritma za generiranje kvadrata na panelima druge fasade

Navedene geometrijske oblike je moguće pretvoriti u zatvorene plohe koje potom možemo koristiti i za vizualizaciju primjera, odnosno možemo pokušati generirati prikaz panela i sjene koju pružaju u uvjetima stvarnog osvjetljenja (slika 9.). Svi algoritmi mogu se jednostavno prilagoditi odnosno doraditi za izradu sličnih geometrijskih oblika, no na sličan način moguće je kreirati i nove oblike panela. Sve to omogućava arhitektima jednostavnu prilagodbu geometrijskih oblika pojedinoj građevini s obzirom na potrebe osvjetljenja no i s obzirom na željenu estetiku konačne fasade. Iako na prvi pogled sva rješenja djeluju geometrijski jednostavno i djeluje da ih je možda moguće brzo napraviti ručno, to nipošto nije slučaj jer primjerice samo druga fasada iz primjera sadrži čak 500 kružnica na cijeloj fasadi, dok primjerice četvrta fasada sadrži više od 8500 kvadrata na cijeloj fasadi i svaku od tih kružnica ili svaki od tih kvadrata je potrebno posebno modificirati sukladno željenom pravilu. Ručna izrada svakog rješenja bi sigurno trajala danima ili čak tjednima te jednostavno nije ekonomski isplativa.

50



Slika 9. Prikaz sjena panela u uvjetima stvarnog osvjetljenja

konačne varijante potrebno je provjeriti tehničku sukladnost rješenja s uvjetima koje je potrebno ispuniti, te postoji mogućnost da će neko od rješenja biti odbačeno iz razloga neispunjavanja tehničkih uvjeta iako rješenje zadovoljava svojim izgledom. Bez obzira na tu mogućnost, promjenom jednog ili više ulaznih parametara možemo jednostavno dobiti novo rješenje koje će zadovoljiti tehničke uvjete. S obzirom da ispunjavanje tehničkih uvjeta ovisi najčešće o dimenziji određenih dijelova panela, moguće je u postojeći algoritam ugraditi novi dio algoritma koji može tu provjeru raditi automatski prilikom izračuna i odmah ukazivati na pro-

blematične dijelove. Na kraju, iako je prikazane primjere fasada moguće izraditi klasičnom geometrijskom konstrukcijom kod koje projektant samostalno nasumično određuje položaj, orijentaciju i oblik svih otvora na svim panelima, postizanje istih takvih rješenja bi u većini slučajeva zahtijevalo dane ili tjedne računanja i konstruiranja samo jedne varijante izgleda fasade, a što je neusporedivo sporije od korištenja predloženog programskog alata. Kad se u obzir uzme potencijalna potreba izrade više varijantnih rješenja što zbog postizanja željene estetike ili zbog ispunjavanja tehničkih uvjeta, navedena razlika u trajanju izrade finalnog rješenja još više dolazi do izražaja i izrada klasičnim „ručnim“ načinom gubi svaki smisao. Sve navedeno dobiva još više na značenju ako u jednadžbu uvrstimo i potrebu za izradom „nasumičnih“ unikatnih rješenja koja bez upotrebe računala jednostavno rečeno nije moguće napraviti u ekonomski isplativom roku.

4. Zaključak

Projektiranje građevinskih elemenata zahtijeva razna tehnička znanja poput znanja o tehničkoj mehanici ili znanja o građevinskim materijalima, no osim tih znanja nužnih za projektiranje sigurne i dugotrajne građevine potrebna su i druga znanja koja će građevini dati unikatni izgled uz ispunjavanje svih tehničkih uvjeta. Unikatni izgled građevine moguće je postići korištenjem raznih kreativnih tehnika kojima je uvijek cilj vizualno isticanje zgrade u odnosu na ostale u njejoj okolini. Iako kreativnih tehnika za postizanje unikatnog izgleda ima iznimno velik broj, neke zahtijevaju kompleksnu izradu građevinskih elemenata, neke su vremenski zahtjevne u postupku projektiranja ili izgradnje (ili oboje), a neke su jednostavno preskupe i ekonomski neisplative. Cilj ovog rada je bio proučiti neke mogućnosti programskog alata za modeliranje opće namjene Rhinoceros 3D i dodatka za vizualno programiranje Grasshopper, te vidjeti je li moguće korištenjem ugrađenog generatora pseudoslučajnih brojeva izraditi jednostavnije građevinske elemente koji bi se međusobno razlikovali i koji bi dali građevini unikatni izgled. Pritom je važno da ti elementi ispune osnovne tehničke uvjete za građevinu i da u što većoj mjeri ispune uvjete jednostavnosti izrade, što manjeg utroška vremena, i financijske isplativosti. S obzirom da se za izradu metalnih fasadnih panela koristi tehnologija rezanja strojnim rezačima koji kao ulazni parametar za svoj rad koriste računalno generiranu grafiku u vektorskom formatu, i da je stroju moguće jednostavno unijeti veliki broj različitih predložaka, izbor Rhinoceros i Grasshopper alata se nametnuo kao logičan odabir jer oba alata koriste vektorski format zapisa prikladan za navedenu svrhu. Upotrebom oba alata izrađen je Grasshopper

algoritam za četiri različita primjera fasadnih panela prikladnih za izradu navedenom tehnologijom, a koja zadovoljavaju sve uvjete vizualne unikatnosti. Algoritmi su izrađeni tako da je jednostavnim mijenjanjem ulaznih parametara moguće dobiti varijantna rješenja koja koriste isti osnovni uzorak no čiji je rezultat zbog korištenja PRNG-a jedinstveni izgled svakog pojedinog panela. Na kraju, iako su sva četiri primjera fasada zadovoljila uvjete vizualne unikatnosti, kao završni korak prije proizvodnje bilo bi potrebno izvršiti detaljniju provjeru njihovih tehničkih karakteristika koje moraju ispuniti kao i svi drugi građevinski elementi. Također, važno je napomenuti da sva četiri rješenja, iako tehnički izvediva tehnologijom strojnog rezanja, imaju drastično različit izgled koji značajno utječe na njihovu izradu navedenom tehnologijom. To se prvenstveno odnosi na duljinu potrebnog rezanja na pojedinom panelu jer duljina rezanja je glavni faktor koji određuje trajanje rada stroja i izravno utječe na cijenu pojedinog panela. Za kvalitetnu ocjenu sva četiri predložena izgleda panela bilo bi potrebno proučiti tehničke karakteristike svakog pojedinog rješenja u kontekstu ispitivanja ispunjavanja propisanih tehničkih uvjeta i u kontekstu tehnološko financijske isplativosti, a što predstavlja dobru smjernicu za nastavak istraživanja ove teme.

Literatura

- [1] S. Jurić, *Generatori slučajnih brojeva*, http://sigurnost.zemris.fer.hr/random/2001_juric/
- [2] I. Urbiha, *Generiranje niza pseudoslučajnih brojeva*, Matematičko fizički list, LXII (2010), 75–82.
- [3] https://rosettacode.org/wiki/Linear_congruential_generator#top-page
- [4] S. K. Park, K. W. Miller, *Random number generators: good ones are hard to find*. Commun, ACM **31**(10), (1988), 1192–1201.
- [5] P. L'Ecuyer, *Tables of linear congruential generators of different sizes and good lattice structure*, Mathematics of computation, **68**(225), (1999) 249–260.
- [6] P. A. W. Lewis, A. S. Goodman, J. M. Miller, *A pseudo-random number generator for the System/360*, IBM Systems Journal, **8**(2), (1969) 136–145.
- [7] F. Aurenhammer, R. Klein, *Voronoi Diagrams, Technical Report 198*, FernUniversität Hagen, Department of Computer Science, Germany, 1996.

- [8] Ž. Gjuranić, *Konike i grafovi nekih polinoma pomoću NURBS krivulja*, KOG **13**, (2009), 51–55.

Boris Uremović
Tehničko veleučilište u Zagrebu, Graditeljski odjel,
10000 Zagreb, Av. V. Holjevca 15, Hrvatska
E-mail adresa: `boris.uremovic@tvz.hr`

Mandi Orlić Bachler
Tehničko veleučilište u Zagrebu, Graditeljski odjel,
10000 Zagreb, Av. V. Holjevca 15, Hrvatska
E-mail adresa: `mandi.orlic@tvz.hr`

Marija Čačić Ptičar
studentica 3. godine preddiplomskog stručnog studija graditeljstva,
Tehničkog veleučilišta u Zagrebu
E-mail adresa: `marija.cacic@tvz.hr`