

GENERATIVNI PRISTOP K UČINKOVITEMU NAČRTOVANJU OBJEKTOV

GENERATIVE APPROACH TO EFFICIENT BUILDING DESIGN

asist. Luka Gradišar, mag. inž. grad.

luka.gradisar@fgg.uni-lj.si

doc. dr. Matevž Dolenc, univ. dipl. inž. grad.

matevz.dolenc@fgg.uni-lj.si

asist. dr. Robert Klinc, univ. dipl. inž. grad.

robert.klinc@fgg.uni-lj.si

prof. dr. Žiga Turk, univ. dipl. inž. grad.

ziga.turk@fgg.uni-lj.si

Univerza v Ljubljani, Fakulteta za gradbeništvo
in geodezijo, Katedra za gradbeno informatiko,
Jamova 2, Ljubljana

Znanstveni članek

UDK 004.414.23:69-5

Povzetek | Pri načrtovanju se velikokrat srečamo s problemi, za katere moramo določiti ustrezno rešitev. Takšne probleme običajno rešujemo iterativno (predpostavimo rešitev, jo analiziramo in preverimo ustreznost, dokler rešitev ne zadosti vsem pogojem in ciljem) na podlagi izkušenj in znanja, kar pa ne deluje pri problemih, s katerimi se soočimo prvič. Alternativen pristop je generativno načrtovanje, kjer se osredotočimo na definiranje problema s pripadajočimi cilji v obliki parametričnega računskega modela, ki ga prepustimo v reševanje algoritmom. Ti na podlagi optimizacijskih metod iščejo ustrezne rešitve glede na podane kriterije in omejitve. Rezultat generativnega postopka je množica rešitev, ki nam z ustrezno analizo pomagajo učinkovito rešiti zastavljen problem. Glavna ideja pristopa je sodelovanje projektanta z inteligentnim orodjem, ki ima komplementarne sposobnosti: pregled velikega števila podatkov, generiranje velikega števila alternativ, analiza množice in iterativno izboljšanje rešitev. Pristop je splošen in zato uporaben pri načrtovanju na vseh področjih gradbeništva. V prispevku se vprašamo o zmogljivosti takega pristopa in če ta vodi k nadomestitvi projektanta v procesu načrtovanja. Tako v prvem delu predstavimo ključne dele generativnega načrtovanja in njihovo delovanje, v drugem delu na primeru uporabimo proces generativnega pristopa za načrtovanje elementov za senčenje, v zadnjem delu pa predstavimo rezultate obeh pristopov.

Ključne besede: generativno načrtovanje, optimizacija, evolucijski algoritmi, računski model, Dynamo, BIM

Summary | When designing, we often face problems for which we must find a solution. We solve such problems iteratively by proposing a solution, analysing it and testing its suitability until the solution satisfies all conditions and objectives. Normally we solve such problems based on experience and knowledge, which does not work for the problems we are confronted with for the first time. An alternative approach is generative design, where we focus on defining a problem with its objectives and constraints in terms of a parametric computational model and leave the solution to the algorithms. The result of the generative process is a set of solutions the analysis of which helps us to solve the problem. The main idea of the approach is the collaboration between the designer and the intelligent tool, which has complementary capabilities, such as sorting large amounts of data, generating many alternatives, analysing the solutions, and incrementally improving

them. The approach is general and can also be applied in other areas of civil engineering. In this paper, we question the capacity of such an approach and whether it aims to replace the designer in the design process. Therefore, in the first part we present the key elements of generative design, in the second part we apply the generative design process for the design of shading elements, and in the last part we present the results and compare them with the usual approach.

Key words: Generative design, optimisation, evolutionary algorithms, computational model, Dynamo, BIM

1 • UVOD

V zadnjih letih je opazen vedno hitrejši tehnološki razvoj, ki se v gradbeništvu odraža v vedno večji uporabi digitalnih orodij. Posledično se delo projektantov drastično spreminja, saj se je zaradi menjave svinčnika in papirja s CAD-orodij proces načrtovanja stavb precej poenostavil in pohitril. Prav tako se z menjavo CAD-orodij z informacijskimi modeli spreminja tudi proces načrtovanja. Vedno manjša je potreba po risarjih, načrtovanje se zliva z dokumentiranjem in je bistveno bolj celovito, povezano in koordinirano. Obenem so orodja

za numerično analizo računalniško podprta, kar bistveno poenostavi in pospeši preverjanje rešitev. Še vedno pa mora rešitve, ki jih računalniki pomagajo analizirati in dokumentirati, poiskati inženir ali arhitekt.

Trenutno se pri projektiranju večinoma še vedno uporablja »klasični« pristop. Projektant mora razumeti problem, njegove kriterije in omejitve, nakar lahko rešitve ustvari glede na svoje znanje, izkušnje, podobne pretekle primere in domišljijo, seveda v odvisnosti od vhodnih pogojev in ciljev. Omejitve tako

določijo, katere rešitve so sploh dopustne, kriteriji pa omogočajo kvalitativne primerjave med njimi. V procesu je projektant tisti, ki (pred)postavi rešitev, računalnik pa pomaga pri njeni predstavitvi in analizi, ne pomaga pa pri ustvarjanju rešitev. Za alternativne pristope nas zanimajo takšni, ki preusmerijo osredotočenost iz direktne sinteze rešitev na definiranje problema z jasnimi zakonitostmi. Zastavljeni problem lahko nato prepustimo v reševanje algoritmom, ki so sposobni generirati in potem analizirati veliko število rešitev. Eden izmed teh pristopov, ki je predstavljen v nadaljevanju, je generativno načrtovanje.

2 • GENERATIVNO NAČRTOVANJE

Generativno načrtovanje je iterativni proces, v katerem se projektant osredotoči na definiranje problema in njegovih ciljev, iskanje rešitev pa prepusti algoritmom. Ti s spreminjanjem parametrov problema generirajo množico različnih rešitev, ki jih analiziramo z namenom razumeti obnašanje problema (Krish, 2011). S tako pridobljenim znanjem lahko popravimo in nadgradimo opis problema ali pa izberemo rešitev, ki zadošča vsem zastavljenim ciljem. S takšnim procesom rešujemo probleme, kjer ima projektant premalo izkušenj in znanja, da bi hitro ustvaril rešitev, ki bi zdržala preverbo omejitev in optimalno izpolnila kriterije.

2.1 Sestavni deli generativnega načrtovanja

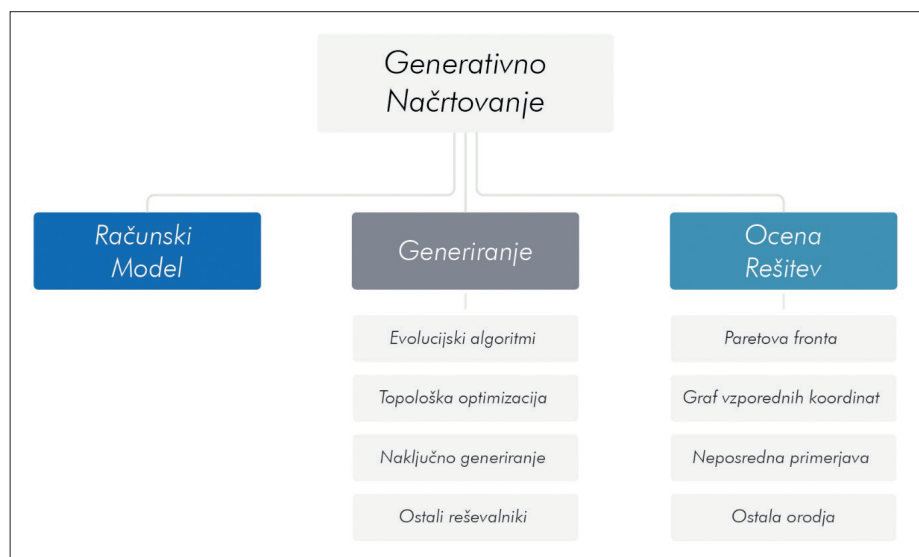
V procesu generativnega načrtovanja lahko identificiramo tri ključne značilnosti, prikazane na sliki 1: računski model, generiranje in oceno rešitev.

Računski model razumemo kot poenostavitev realnega inženirskega problema na model, ki ga računalnik lahko prebere, analizira in spreminja. Izdelava takšnega modela je zahtevna, saj je treba problem zapisati s skupkom pravil in enačb, ki pravilno opišejo njegovo

obnašanje, poleg tega pa mora biti opis parametričen, saj se s parametri spreminjajo zasnova in njene lastnosti (Stasiuk, 2018). Pri vseh parametričnih modelih nas zanimajo tudi njihove lastnosti, na podlagi katerih lahko ocenimo uspešnost zasnove, kot so npr. cena,

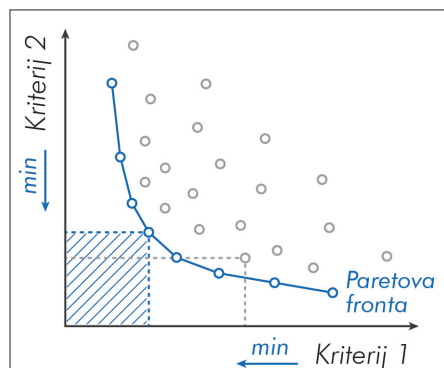
prostornina, učinkovitost, nosilnost, stabilnost in podobni opisi, primerni za primerjavo generiranih zasnov med seboj (Caldas, 2002). Običajno se računski modeli zaradi enkratnosti problema programira samostojno, obstajajo pa programske rešitve za nekatere ponovljive probleme.

Generiranje namesto projektanta izvaja »računalnik« oziroma algoritem, ki avtomatizira parametrično iskanje ter testiranje možnih



Slika 1 • Ključne značilnosti generativnega načrtovanja.

rešitev problema, vendar pa je kvaliteta le-teh pogojena z natančnostjo računskega modela, pripravljenega s strani projektanta. Ti algoritmi



Slika 2 • Simbolični primer Paretove fronte, kjer želimo minimizirati dva kriterija.

delujejo tako, da preko računskega modela spreminjajo vhodne spremenljivke in pri tem spremljajo izhodne vrednosti, s pomočjo katerih ocenijo uspešnost rešitve, nato pa preko več iteracij poiščejo najbolj primerne glede na zastavljene cilje. Zaradi velikega števila spremenljivk in posledično velikega prostora rešitev se običajno uporabljajo optimizacijski algoritmi, najpogostejše evolucijski algoritmi (Johan, 2019) ter algoritmi za topološko optimizacijo v kombinaciji z evolucijskimi (Ravi, 2018). Poleg tega se uporabljajo tudi algoritmi, ki generirajo rešitve po določenih pravilih kombiniranja vhodnih spremenljivk, kot je vektorski produkt ali naključna kombinacija.

Pri oceni rešitev je treba pregledati generirane rešitve, saj je, odvisno od zasnove problema in ciljev, lahko teh zelo veliko, med njimi pa je treba izbrati eno končno rešitev. To pomeni, da je treba oceniti množico rešitev, za kar se uporabijo orodja za analizo podatkov, kot so npr. Paretova fronta, graf vzporednih koordinat in neposredna primerjava, z namenom identificiranja optimalnih rešitev in ugotavljanja vpliva vhodnih spremenljivk na obnašanje modela in na njegove rezultate. S tako pridobljenim znanjem lahko nato nadgradimo računski model ali izberemo rešitev zadanega problema.

Ker se v generativnem načrtovanju pogosto srečujemo z več kot enim ciljem oziroma kriterijem, po katerem ovrednotimo rešitev, obstaja več kot ena optimalna rešitev. Take rešitve imenujemo Pareto optimalne rešitve in jih identificiramo s Pareto fronto (Deb, 2002). Zanje je značilno, da ne obstaja rešitev, ki bi bila boljša po vseh kriterijih hkrati (prikazano na sliki 2 z modrim območjem). Obstajajo pa rešitve, ki so lahko boljše po posameznem kriteriju (označene s sivo barvo na sliki 2), vendar te niso optimalne pri večkriterijski optimizaciji (Allen, 2018).

Da ugotovimo, kako vhodne spremenljivke vplivajo na izhodne, nam v pomoč pride graf vzporednih koordinat, ki prikaže medsebojne povezave spremenljivk vsake rešitve. Poleg tega lahko na grafu dodatno omejimo vrednosti spremenljivk in s tem zmanjšamo množico rešitev na manjši vzorec, ki ga je

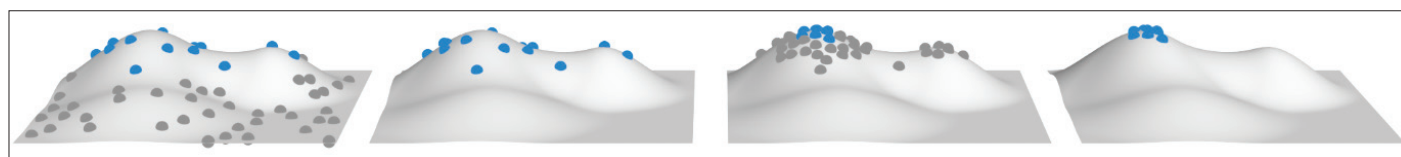
lažje obvladovati. V kombinaciji z neposredno primerjavo se izbrane rešitve prikažejo še grafično druga ob drugi in s tem omogočijo hitro primerjavo ter učinkovito iskanje ustrezne rešitve.

2.2 Evolucijski algoritmi

Pogosto uporabljeni algoritmi za iskanje in generiranje rešitev pri generativnem načrtovanju so evolucijski algoritmi. Ti delujejo po principu evolucije, kjer uspešni člani populacije preživijo in se reproducirajo, pri tem pa se z vsako novo generacijo člani izboljšujejo, dokler ne dosežejo vrha. Eden izmed evolucijskih algoritmov je genetski algoritem, katerega postopek je naslednji (Rutten, 2010):

1. naključno izbere prvo generacijo v prostoru rešitev;
2. vse člane razvrsti po uspešnosti in obdrži delež najbolj uspešnih;
3. ustvari novo generacijo s križanjem in mutacijo;
4. ponavlja, dokler z novo generacijo ni znatne izboljšave.

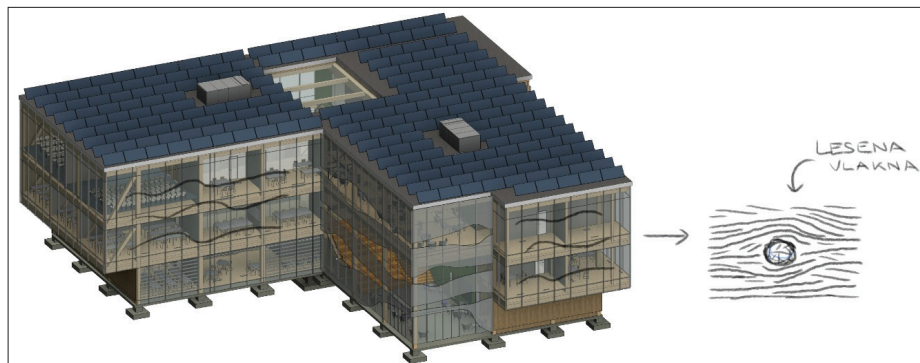
Križanje deluje na podlagi menjave genov oziroma vrednosti parametrov, mutacija pa naključno priredi vrednosti teh parametrov in s tem prepreči zatik populacije v lokalnem optimumu in zagotovi konvergenco h globalnemu (Guid, 2015). Genetski algoritem tako s pomočjo populacije hitro poišče globalni ekstrem (maksimum ali minimum) in lahko privede do zanimivih in manj vidnih rešitev, ki rešijo zastavljen problem (Konak, 2006).



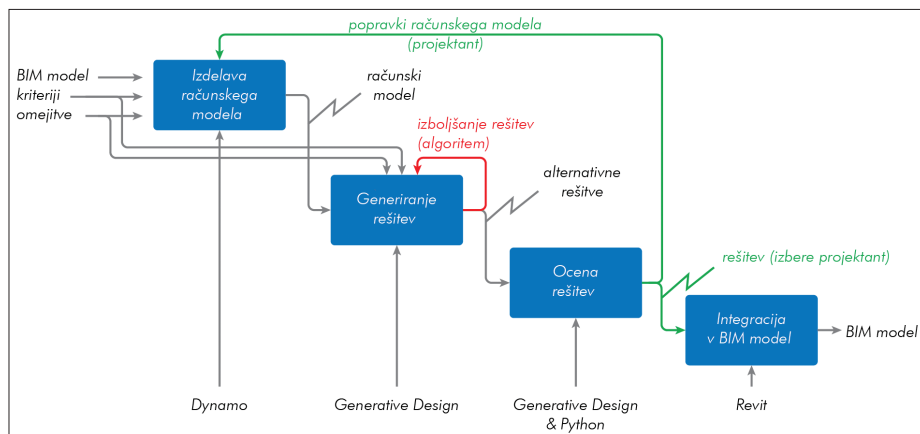
Slika 3 • Prikaz iskanja globalnega vrha z evolucijskimi algoritmi na preprostem primeru. Iz leve proti desne je viden postopek iterativnega izboljšanja rešitev.

3 • PRIKAZ UPORABE NA PRIMERU

Primerjavo generativnega načrtovanja z običajnim pristopom smo izvedli na primeru načrtovanja elementov za senčenje stavbe. Zaradi vedno večjega poudarka na energetski učinkovitosti je bila želja poiskati učinkovito zasnovo senčenja velikih steklenih površin z idejo nepremičnih elementov v obliki lesenih vlaken (slika 4), ki bi poudarili leseno konstrukcijo stavbe in pri tem učinkovito senčili v vseh letnih časih. Z uporabo generativnega pristopa je bil cilj poiskati optimalno obliko elementov za senčenje, ki bi pri čim cenejši izgradnji učinkovito senčila in poudarila kompleksen videz fasade.



Slika 4 • Ideja nepremičnih elementov za senčenje v obliki lesenih vlaken.



Slika 5 • IDEF0-diagram generativnega načrtovanja na predstavljenem primeru.

Za uporabo generativnega pristopa je bilo treba določiti okolje za izgradnjo računskega modela, orodje za generiranje rešitev in za vrednotenje rezultatov. V ta namen smo izbrali orodja Autodesk Revit, Dynamo, programski jezik Python in dodatek Generative Design (v času opravljanja raziskave se je imenoval še Project Refinery), ki je od različice 2021 dalje že privzeto vgrajen v program Revit (Autodesk, 2020). V programu Revit je bil zgrajen informacijski model stavbe, ki je predstavljal izhodišče, na katerem se je zasnovalo in nato dodalo model elementov za senčenje. Zastavljenega kompleksnega modela ni možno modelirati ročno, zato se je za računsko modeliranje uporabil Dynamo, kjer preko vizualnega programiranja z vsakim delčkom kode gradimo model ali pa upravljamo informacije (The Dynamo Primer, 2020). Za generiranje rešitev se je uporabil dodatek Project Refinery, ki vsebuje različne algoritme za generiranje rešitev in vmesnik za pregled rezultatov (Generative Design Primer, 2020). Rezultati so se urejali s pomočjo programskega jezika Python, kjer sta se uporabila Paretova fronta in graf vzporednih koordinat.

3.1 Računski model

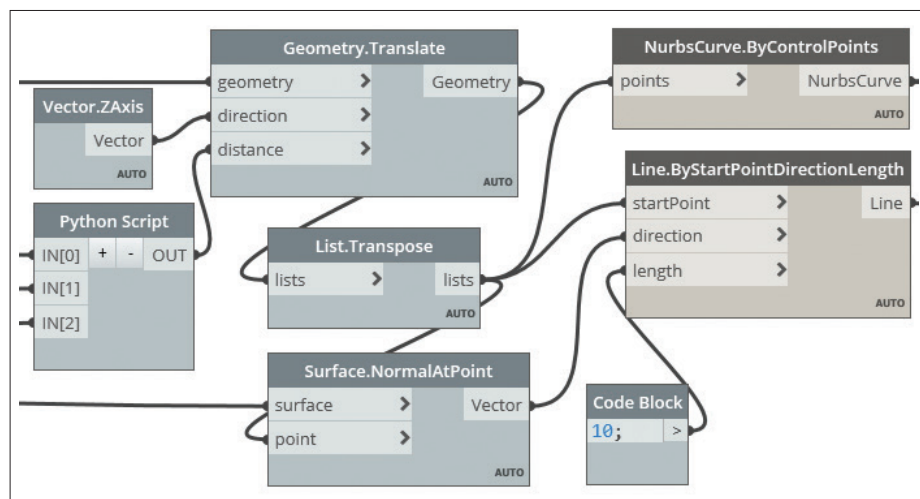
V našem primeru je računski model predstavljala skripta v okolju Dynamo, v katerem se je s povezovanjem blokov kode (slika 6) sestavilo, definiralo obnašanje in analiziralo elemente za senčenje. Kot izhodišče se je iz informacijskega modela preneslo ploskve zastekljenih površin, njihovo lego in orientacijo ter smeri sonca v različnih letnih časih. Površine so se razdelile na horizontalne elemente, ki se jih je nato preko oblikovnih krivulj, trigonometričnih funkcij in z uvedbo manjših odprtin prevedlo v kompleksnejšo obliko, prikazano na sliki 7.

Parametri računskega modela, ki so spreminjali zasnovo elementov za senčenje, so bili: število horizontalnih elementov, največja širina elementov, lega in velikost odprtin ter

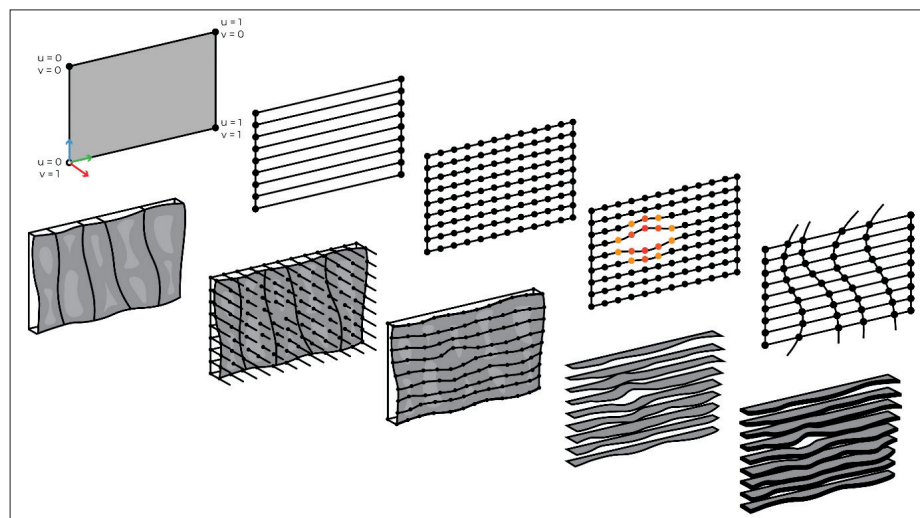
interpolacijske točke oblikovnih krivulj, ki so definirale valovito geometrijo elementov. Vsi ti parametri so imeli vpliv na obliko elementov in posledično na njihovo učinkovitost senčenja. Za oceno te učinkovitosti smo pripravili analizo senčenja, kjer se je s pomočjo simuliranja sončnih žarkov preverilo, koliko jih preide v notranjost oziroma koliko jih elementi ustavijo (prikazano na sliki 8 z modro in rdečo barvo). Za dodatni deterministični kriterij se je iz modela izvleklo skupno porabo materiala za analizo stroškov.

3.2 Generiranje rešitev

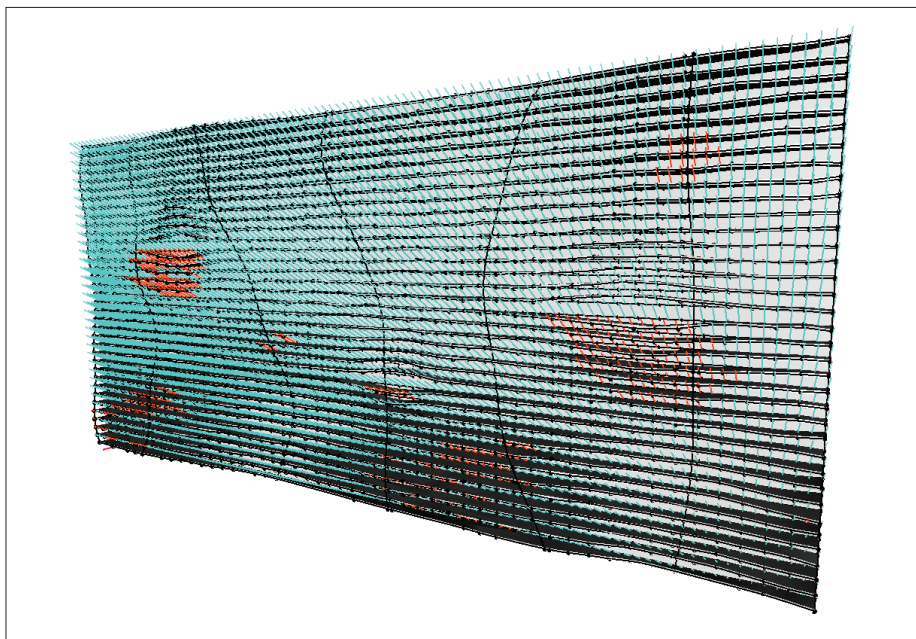
Z zgrajenim računskim modelom je iskalnik rešitev, ki ga vsebuje dodatek Project Refinery, generiral rešitve s spreminjanjem vhodnih spremenljivk na podlagi optimizacije z genetskimi algoritmi. Zaradi želje po hitrejših izračunih sta se za vhodne spremenljivke



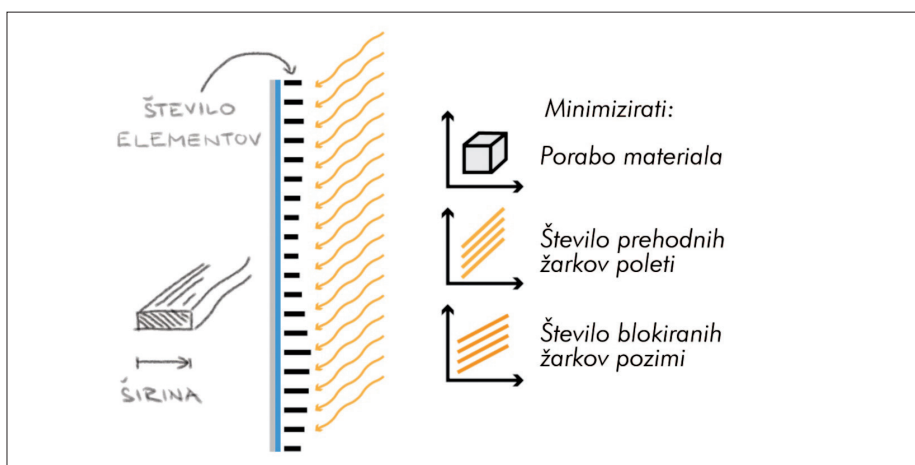
Slika 6 • Primer blokov kode, s povezovanjem katerih se je zgradil računski model elementov za senčenje.



Slika 7 • Ilustracija izgradnje računskega modela v programu Dynamo.



Slika 8 • Model v okolju Dynamo s prikazano analizo senčenja.



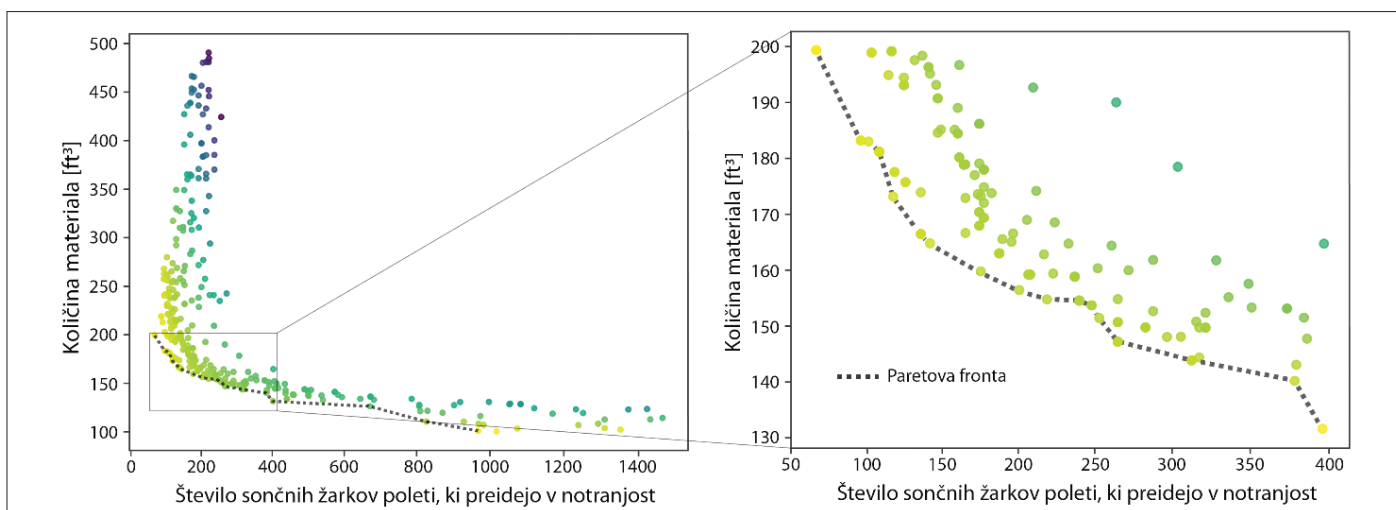
Slika 9 • Vhodne in izhodne spremenljivke za optimizacijo elementov za senčenje.

izbrala le število horizontalnih elementov in koeficient širine elementov, saj se je preko iterativnega procesa izdelave računskega modela in rezultatov analiz ugotovilo, da imata ti dve največji vpliv na vrednosti izhodnih spremenljivk. S tem se je število vseh možnih kombinacij zmanjšalo na le 18.200. Za izhodne spremenljivke, na podlagi katerih so se vrednotile rešitve, so bile izbrane poraba materiala, število prehodnih žarkov poleti in število blokiranih žarkov pozimi (slika 9), s čimer se je številčno opredelila učinkovitost senčenja. Iskale so se rešitve, ki so imele čim manjše vrednosti vseh treh izhodnih spremenljivk. Za dodatni kriterij se je spremljalo še videz elementov, vendar takega kriterija ni bilo mogoče numerično opredeliti, zato se je ocenil subjektivno.

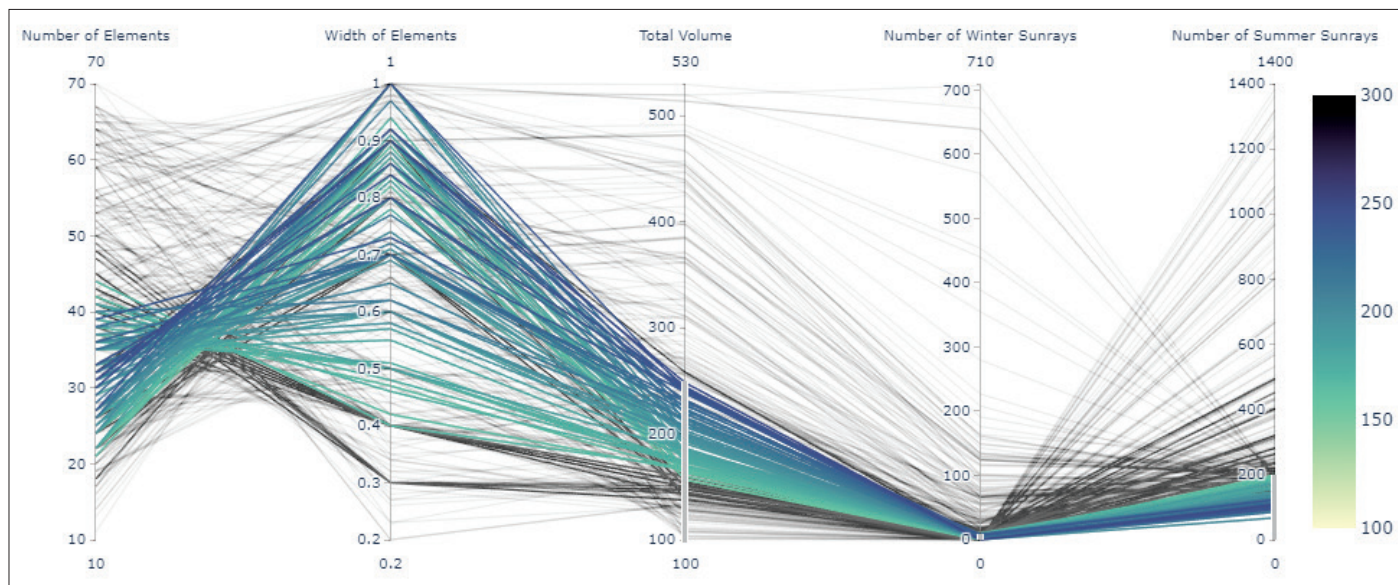
3.3 Ocena rešitev

Množica rešitev je bila dobljena v več korakih z različnimi nastavitvami natančnosti optimizacijskega algoritma, kar nam je vrnilo vzorec 573 raznolikih rešitev. Tako število podatkov težko pregledamo individualno, zato se je z uporabo Paretove fronte, grafom vzporednih koordinat in neposredno primerjavo analizirala množica rešitev in pri tem ugotavljalo delovanje modela.

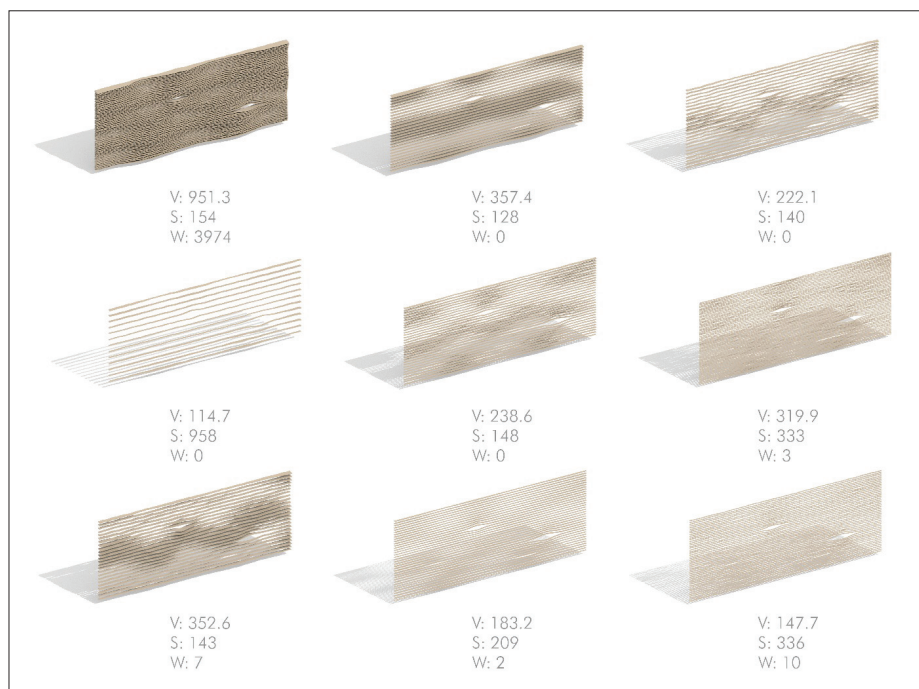
Paretova fronta se je uporabila za identifikacijo optimalnih rešitev (slika 10), saj smo imeli več kriterijev, po katerih smo ocenjevali različne rešitve. Ker nas je poleg obravnavanih kriterijev zanimal tudi videz elementov, ki ga ni bilo možno ovrednotiti v računskem modelu, nas niso zanimale zgolj Pareto optimalne rešitve, temveč tudi takšne z zanimivo obliko. Te rešitve, ležeče na Paretovi fronti, so nam podale referenčne vrednosti kriterijev, s katerimi smo



Slika 10 • Paretova fronta, zgrajena na rezultatih iz optimizacije.



Slika 11 • Omejevanje rešitev z grafom vzporednih koordinat (barve predstavljajo spremembo količine materiala).



Slika 12 • Neposredna primerjava alternativnih rešitev.

ocenili, koliko daleč od optimalnih se nahajajo ostale rešitve, ki smo jih obravnavali v procesu ocenjevanja.

Množica rešitev se je prikazala še na grafu vzporednih koordinat, kjer so povezave med lastnostmi posameznih rešitev pokazale vpliv števila elementov in koeficienta širine elementov na izhodne spremenljivke. Raztros spremenljivk je pokazal večji vpliv števila elementov na učinkovitost senčenja, saj ima ta manjši raztros kot širina elementov. Poleg tega so se na grafu dodatno

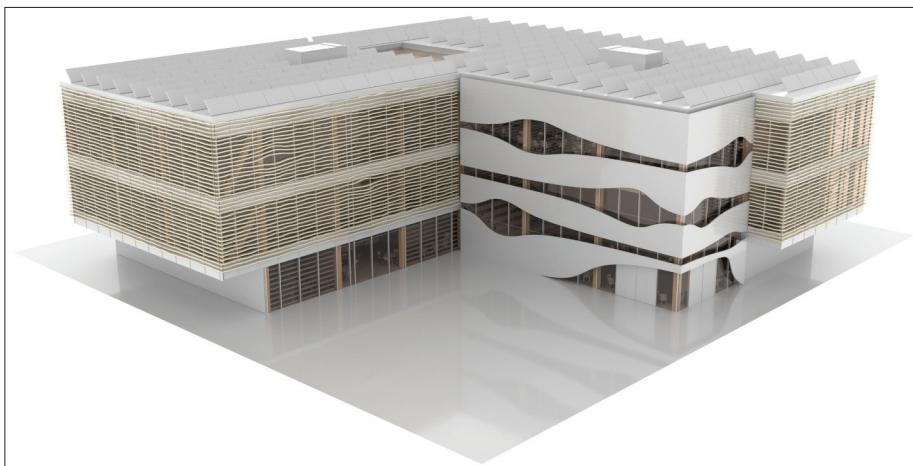
omejili vrednosti spremenljivk in s tem število rešitev na manjši vzorec, prikazan na sliki 11. Spreminjanje mej je vrnilo različne vzorce, ki so se z neposredno primerjavo (slika 12) primerjali še po videzu, kar nam je omogočilo hiter pregled več ustreznih alternativnih rešitev.

3.4 Rešitev problema

Med procesom generativnega načrtovanja se je računski model postopoma nadgrajeval, pri tem pa smo spremljali rezultate, dokler

se ni našla ustrezna rešitev. Ugotovili smo, pri kakšnih pogojih optimalno rešijo problem, kako parametri vplivajo na obliko in pripadajoče lastnosti, in s tem dobili vpogled v zastavljeni problem. S tako pridobljenim znanjem smo izbrali rešitev, ki je zadostila vsem pogojem in ciljem, in jo dodali informacijskemu modelu, kar je prikazano na sliki 13. Izbira rešitve je bila še vedno odvisna od projektantov, saj je problem zajemal več kriterijev, med njimi tudi subjektivna ocena videza fasade, ki je pametni računalniki še ne znajo ovrednotiti.

Pred uporabo generativnega pristopa smo elemente za senčenje zasnovali z običajnim pristopom, kjer se je oblika spreminjala ročno, nato smo analizirali senčenje in ocenili ustreznost podane rešitve. V tem pristopu je bilo težje oceniti ustreznost rešitve, saj nismo imeli referenčnih vrednosti, in se ni vedelo, ali se lahko rešitev še izboljša. Z uporabo generativnega načrtovanja smo ugotovili, da obstaja več takih rešitev, ki jih je bilo mogoče med seboj primerjati in poiskati najbolj ustrezno. Primerjava obeh pristopov je pokazala, da je rezultat generativnega pristopa imel 307% manjšo porabo materiala, enako učinkovitost poleti in bil učinkovit tudi pozimi. V primeru ročnega spreminjanja se je učinkovitost pozimi zanemarljivo, saj je bilo težje spremljati vse cilje.



Slika 13 • Vizualizacija informacijskega modela stavbe z dodanimi elementi za senčenje.



Slika 14 • Vizualizacija končne rešitve elementov za senčenje.

4 • SKLEP

V prispevku smo predstavili ključne značilnosti generativnega načrtovanja in prikazali njihovo delovanje na primeru. V tem pristopu se namesto iskanja posamične rešitve osredotočimo na definiranje problema z vsemi pogoji in cilji, nakar načrtovanje prepustimo algoritmom. Ker generativno načrtovanje avtomatizira sintetiziranje, analizo in večji del ocene rešitev, predstavlja zmožljiv pristop k načrtovanju.

Na primeru zasnove elementov za senčenje se je iskala rešitev z običajnim pristopom in nato z generativnim načrtovanjem. Generativni pristop je podal rešitev, ki je učinkovito senčila in pri tem imela 307 % manjšo porabo materiala kot rezultat običajnega pristopa. Ko smo se prvič soočili z novim

problemom, je bilo težko predpostaviti rešitev, ki bi zadostila vsem ciljem. Potrebno je bilo iterativno iskanje, ki pa je lahko zamudno, saj pot do optimalne rešitve ni enosmerna. Tukaj je generativno načrtovanje v prednosti, saj avtomatizira iskanje optimalnih rešitev in zmanjša število tistih, med katerimi se izbere končna.

Zmožljivost procesa generativnega načrtovanja se kaže predvsem pri reševanju novih problemov, kjer ni pričakovati, da bi projektant ugotovil rešitev, ki bi bila blizu optimalne. Preko definirane problema se namesto iskanja posamične rešitve ustvari množica rešitev, kar nam poda vpogled v razumevanje problema. Vendarle ta pristop zahteva znanje in čas za izdelavo računskega modela, ki si

ga včasih ne moremo privoščiti pri projektih. Prednost pristopa se pojavi šele pri večkratni uporabi računskega modela, saj ga je možno hitro prilagoditi za podobne primere. Pristop je splošen in zato uporaben tudi kot pomoč pri načrtovanju na drugih področjih gradbeništva, kjer lahko problem opišemo z računskim modelom.

Pri uporabi takega procesa smo se tudi vprašali, ali generativni pristop nadomešča projektanta v procesu načrtovanja. Trenutni odgovor je ne, saj je projektant še vedno odgovoren za dva od treh delov procesa, »računalnik« prevzame le pametno iskanje rešitev zastavljenega problema, projektant pa mora še vedno definirati problem in izbrati končno rešitev.

5 • ZAHVALA

Iskrena zahvala Javni agenciji za raziskovalno dejavnost Republike Slovenije, ki je v okviru programa Mladi raziskovalci financirala prikazane raziskave.

6 • LITERATURA

- Allen, M., Python for healthcare modelling and data science, pareto front, <https://pythonhealthcare.org/tag/pareto-front/>, 2018.
- Autodesk, Generative Design in Revit now available, <https://blogs.autodesk.com/revit/2020/04/08/generative-design-in-revit/>, 2020.
- Caldas, L. G., Norford, L. K., A design optimization tool based on a genetic algorithm, Automation in Construction 11 (2): 173-184, 2002.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computing 6 (2): 182-197, 2002.
- Generative Design Primer, Generative Design for Revit and Dynamo, https://www.generativedesign.org/01-introduction/01-05_gd-for-revit/, 2020.
- Guid, N., Strnad, D., Umetna inteligenca, Maribor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, 2015.
- Johan, R., Chernyavsky, M., Fabbri, A., et al., Building intelligence through generative design, Proceedings of the 24th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Wellington, New Zealand, 2019.
- Konak, A., Coit, D. W., Smith, A., Multi-objective optimization using genetic algorithms: A tutorial, Reliability Engineering and System Safety 91: 992-1007, 2006.
- Krish, S., A practical generative design method, Computer-Aided Design 43 (1): 88-100, 2011.
- Ravi, A., What Generative Design Is and Why It's the Future of Manufacturing, <https://www.newequipment.com/research-and-development/article/22059780/what-generative-design-is-and-why-its-the-future-of-manufacturing>, 2018.
- Rutten, D., Evolutionary Principles applied to Problem Solving, Grasshopper blogs, <https://www.grasshopper3d.com/profiles/blogs/evolutionary-principles>, 2010.
- Stasiuk, D., Design Modeling Terminology, <https://archinate.files.wordpress.com/2018/06/dstasiuk-design-modeling-terminology1.pdf>, 2018.
- The Dynamo Primer, What is Dynamo, https://primer.dynamobim.org/01_Introduction/1-2_what_is_dynamo.html, 2020.