



Univerza v Mariboru

Fakulteta za naravoslovje  
in matematiko



D I A N O I A

REVIIA ZA UPORABO NARAVOSLOVNO-MATEMATIČNIH ZNANOSTI

<b>ISSN</b>	<b>2536-3565</b>
<b>Naslov publikacije/Title</b>	<b>DIANOIA</b> , revija za uporabo naravoslovnih in matematičnih znanosti <b>DIANOIA</b> , journal for applications of natural and mathematical sciences
<b>Letnik/Volume</b>	<b>10</b>
<b>Leto/Year</b>	<b>2026 (april)</b>
<b>Številka/Number</b>	<b>1</b>
<b>Založnik in izdajatelj/ Published &amp; Issued by</b>	Univerzitetna založba Univerze v Mariboru, Slomškov trg 15, 2000 Maribor, Slovenija, <a href="http://press.um.si/">http://press.um.si/</a> , <a href="mailto:zalozba@um.si">zalozba@um.si</a>
<b>Uredništvo/Editorial board</b>	<i>odgovorni urednik/editor in chief</i> Mitja Slavinec <i>glavni urednik/executive editor</i> Drago Bokal <i>izvršna urednica/managing editor</i> Janja Jerebic <i>urednici za področje biologije/editors for biological sciences</i> Nina Šajna, Sonja Škornik <i>urednik za področje didaktike/editor for didactical sciences</i> Samo Repolusk <i>urednika za področje fizike/editors for physical sciences</i> Robert Repnik, Aleš Fajmut <i>urednika za področje matematike/editors for mathematical sciences</i> Igor Pesek, Janja Jerebic <i>urednik za področje tehnike/editor for technical sciences</i> Mateja Ploj Vrtič <i>tehnična urednica/technical editor</i> Špela Kajzer
<b>Mednarodni uredniški svet/ International advisory board</b>	Igor Emri (Fakulteta za strojništvo Univerze v Ljubljani, član SAZU), Matej Brešar (FNM, član SAZU), Sergey Pasechnik (Državna fakulteta v Moskvi), Vlad Popa-Nita (Fakulteta za fiziko Univerze v Bukarešti), Blaž Zmazek (FNM), Samo Kralj (FNM), Franci Janžekovič (FNM), Nataša Vaupotič (FNM), Mitja Kaligarič (FNM), Boris Aberšek (FNM), Andrej Šorgo (FNM), Bojan Mohar (Simon Fraser University, Vancouver), Matjaž Perc (FNM), Ivica Aviani (Naravoslovno matematična fakulteta Split), Fahriye Altınay (Univerza v Nikoziji), Andreas M. Hinz (Univerza Ludwig-Maximilians, München)
<b>Oblikovanje/Design</b>	Amadeja Bratuša
<b>Lektoriranje/Proofreading</b>	Ljudmila Bokal
<b>Sedež uredništva/Address</b>	FNM UM, Koroška cesta 160, 2000 Maribor
<b>e-mail</b>	<a href="mailto:dianoia@um.si">dianoia@um.si</a>
<b>internet/web</b>	<a href="http://www.fnm.um.si">www.fnm.um.si</a>
<b>Tisk/Printed by</b>	FNM UM
<b>Leto izida/Year</b>	2026
<b>Datum natisa/Published</b>	2026
<b>Naklada/Nr. of Copies</b>	50 izvodov

Revija izhaja dvakrat letno, predvidoma aprila in septembra.

## Kazalo / Table of Contents

Devet let potovanj po Dianoi znanja Nine years of traveling through the Dianoia of knowledge Mitja Slavinec	5
Medpredmetno povezovanje s pomočjo terenskega dela v okviru pouka naravoslovja Interdisciplinary Integration Using Fieldwork in Natural Science Lesson Kaja Bračič, Katja Stanič, Andreja Špernjak	9
Akademija paralelnega algoritma razveji in omeji za 0/1 nahrbtnik in delo s sistemom Singularity Guide of parallel branch and bound algorithm for the 0/1 knapsack problem and work with the Singularity system Nik Hrastnik , Aleksander Bajc	33
Hibridno modeliranje hrapavosti 3D tiskanih kovin Hybrid modelling roughness of 3D printing metal Matej Babič, Roman Šturm	53



# Devet let potovanj po Dianoi znanja / Nine years of traveling through the Dianoia of knowledge

Mitja Slavinec

*Univerza v Mariboru, Fakulteta za naravoslovje in matematiko, Koroška cesta 160, 2000 Maribor, Slovenija*

Pred nami je nova številka revije Dianoia, ki se v prostore Fakultete za naravoslovje in matematiko (FNM) ne vrača le kot redni letni pregled strokovnega dela, temveč kot simbol vztrajnosti, predanosti in vizije. Vsaka številka prinaša zgodbe o raziskovanju, radovednosti in skupnem ustvarjanju znanja, ki presega meje posameznih učilnic in laboratorijev. Čeprav devet let v zgodovini znanosti morda ne zveni kot dolga doba, je v življenju mlade fakultete in njenih študentov to obdobje, v katerem so dozorele številne generacije, se utrdile ideje in oblikovale poti, po katerih danes samozavestno stopamo naprej.

Spomini me popeljejo devet let nazaj, ko smo s prvimi pobudami snovali temelje te revije. V tistem času je bilo veliko navdušenja, pa tudi vprašanj in izzivov, ki spremljajo vsako novo pobudo. Želeli smo ustvariti spodbudno okolje, v katerem bi našim najbolj nadobudnim študentom in njihovim mentorjem omogočili prostor za objave ter jih spodbudili k razmišljanju onkraj obveznih študijskih nalog. Namen revije nikoli ni bil zgolj raziskovalne narave; želeli smo ustvariti poligon, kjer se študenti naučijo veščin pisanja strokovnih člankov in se postopoma vpeljejo v svet strokovne publicistike. Prav zato je revija že od samega začetka postavljala visoke standarde – tako pri oblikovno-tehničnih zahtevah kot pri vsebinskem nivoju, saj verjamemo, da se le s soočanjem z resničnimi zahtevami akademskega pisanja oblikujejo vrhunski strokovnjaki.

Na tej poti je bila ključna vloga prof. dr. Draga Bokala. Iskreno sem mu hvaležen, ker je verjel v prvotno pobudo in prevzel odgovorno nalogo urejanja revije. Skupaj s svojo ekipo sodelavcev to delo opravlja strokovno, predano in zelo uspešno, s čimer je v reviji in na fakulteti pustil neizbrisan pečat. Moja zahvala gre tudi obema strokovnima organoma revije, celotnemu uredništvu s področnimi uredniki in mednarodnemu uredniškemu svetu za njuno neomajno budnost nad kakovostjo ter seveda vsem avtorjem. Brez vaših prispevkov revije ne bi bilo. Številke govorijo same zase: za nami je 9 letnikov, 18 številok in skupno 107 objavljenih besedil (89 strokovnih člankov in 18 uvodnikov), ki tvorijo dragocen arhiv znanja naše hiše in dokazujejo vztrajnost ter ustvarjalno energijo vseh sodelujočih.

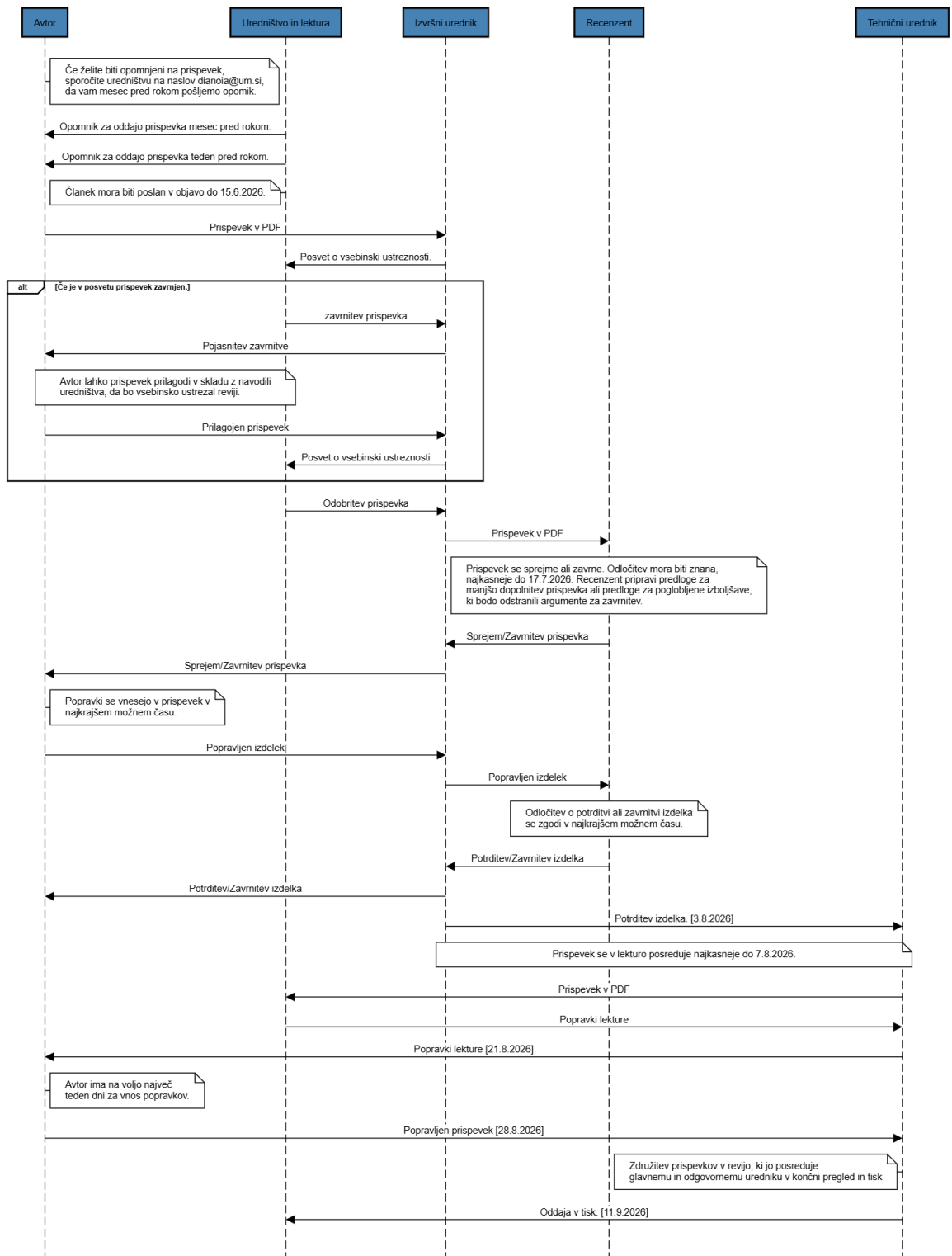
Čeprav devetica še ni povsem okrogla obletnica revije, je pričujoča številka kljub temu jubilejna in izjemno svečana. Izhaja namreč ob 20-letnici Fakultete za naravoslovje in

matematiko. Ta obletnica ni le priložnost za pogled nazaj, temveč tudi trenutek za razmislek o prihodnosti in o tem, kakšno znanstveno skupnost želimo graditi v naslednjih desetletjih. Že od začetka smo termine izdaj zastavili premišljeno – tako, da ena številka izide ob obletnici FNM, druga pa v jesenskem času. Ta povezanost revije z rojstnim dnevom naše fakultete poudarja dejstvo, da je *Dianoia* njen živi utrip, njen odsev in hkrati njen glas.

Naj bo pogled v prihodnost poln optimizma in ustvarjalnega poguma. Misija *Dianoie* ostaja nespremenjena: biti most med radovednostjo študenta in modrostjo mentorja, biti prostor, kjer se drzne ideje prelijejo v urejene vrstice, in biti dokaz, da naša fakulteta ne le poučuje, temveč tudi ustvarja novo znanje. Naj revija v naslednjih desetletjih še naprej raste skupaj z našo fakulteto, utrjuje svojo prepoznavnost in ostaja svetilnik vsem, ki iščejo svojo pot v svetu naravoslovja in matematike.

Srečno, *Dianoia*, in iskrene čestitke ob jubileju naše fakultete.

**prof. dr. Mitja Slavinec**  
odgovorni urednik



Slika 1: Proces izdaje naslednje številke Dianoia. Vir:lasten



# Medpredmetno povezovanje s pomočjo terenskega dela v okviru pouka naravoslovja

## Interdisciplinary Integration Using Fieldwork in Natural Science Lesson

Kaja Bračič<sup>1</sup>, Katja Stanič<sup>2</sup>, Andreja Špernjak<sup>2</sup>

<sup>1</sup>Osnovna šola Martina Konšaka Maribor, Prekmurska ulica 67, 2000 Maribor

<sup>2</sup>Univerza v Mariboru, Fakulteta za naravoslovje in matematiko, Koroška cesta 160, 2000 Maribor, Slovenija

---

### Povzetek

Poučevanje bioloških vsebin se v slovenskih šolah praviloma izvaja v učilnicah. S poučevanjem v naravnem okolju lahko pri učencih izboljšamo razumevanje učne vsebine, saj aktiviramo povezovanje abstraktnih pojmov s konkretnimi primeri, povečamo učno motivacijo in zanimanje za naravoslovje ter prispevamo k razvoju okoljske ozaveščenosti. Terensko delo se večinoma izvaja v obliki naravoslovnih dni, redkeje pa se učitelji odločijo izvesti učno uro naravoslovnih vsebin v naravi. Da bi povečali izvedbo krajših učnih ur zunaj učilnice, smo oblikovali uro naravoslovja v naravi, ki traja 60 minut in vključuje vsebine o snoveh in rastlinah, ki jih učenci obravnavajo pri predmetu naravoslovje v 6. razredu. Sestavljena je iz sedmih vsebinsko različnih dejavnosti, ki vključujejo medpredmetno povezovanje, metodo didaktične igre, uporabo merilnih pripomočkov in določevalnih ključev, izvajanje preprostih poskusov ter mikroskopiranje. Prispevek je namenjen učiteljem kot idejni pripomoček izvedbe ure naravoslovja v naravi, kjer je poudarek na sodelovalnem delu ter medpredmetnem povezovanju za razvoj različnih spretnosti, znanja, vrednot in odnosov. Gradivo je bilo pripravljeno za izvedbo v parku pred Fakulteto za naravoslovje in matematiko in je bilo tudi pilotno preizkušeno z učenci 6. razreda osnovne šole. Na podlagi izvedbe smo analizirali potek aktivnosti ter odzive učencev, kar je omogočilo nadaljnjo nadgradnjo in prilagoditev dejavnosti za širšo uporabo.

*Ključne besede: biologija, kemija, medpredmetno povezovanje, odnos do narave, terensko delo*

### Abstract

Teaching biological content in primary schools is mostly conducted in classrooms. Teaching in the natural environment can enhance students' understanding by connecting abstract concepts with real-life examples, increasing motivation and interest in science, and contributing to the development of environmental awareness. Fieldwork is mostly carried out during science-themed days, while teachers less frequently choose to conduct regular biology lessons outdoors. For this purpose, we designed a 60-minute natural science lesson in nature intended for 6th-grade students. The lesson consists of seven activities with diverse content, integrating cross-curricular connections, didactic games, the use of measuring tools and identification keys, experiments, and microscopy. This contribution is intended for teachers who recognize the importance of outdoor learning, collaborative work, and interdisciplinary teaching in fostering skills, knowledge, values, and relationships associated with this form of education. The material was prepared for implementation in the park in front of the Faculty of Natural Sciences and Mathematics and was also pilot tested with 6th grade elementary school students. Based on the implementation, we analyzed the course of the activity and the students' responses, which enabled further upgrading and adaptation of the activity for wider use.

*Key words: biology, chemistry, cross-curricular connections, attitude towards nature, fieldwork*

---

## 1 UVOD

Biologija preučuje živa bitja, zato pri njenem poučevanju izbiramo aktivne načine učenja in ustrezne didaktične pristope [1], kot so laboratorijsko delo, delo z živimi organizmi in terensko delo. Terensko delo je osnovna metoda nekaterih bioloških ved, kot so etologija, evolucija, ekologija, sistematika in naravovarstvo, zato je smiselno, da ga vključujemo tudi v proces poučevanja. Fleischer in sod. [2] poudarjajo pomen terenskega dela pri poučevanju biologije, saj menijo, da ima v knjigah in na računalnikih abstraktno znanje pogosto prednost pred

---

*E-mail naslov/i:* kaja.bracic1@student.um.si (Kaja Bračič), katja.stanic@student.um.si (Katja Stanič), andreja.spernjak@um.si (Andreja Špernjak)

realnim. Šele z neposrednim stikom z naravo učenci izkusijo naučene pojme, o njih kritično razmišljajo ter razvijajo nove koncepte.

Terensko delo vsebuje metodo, pri kateri učenci pridobijo bogate izobraževalne izkušnje, ki jih v učilnici ni mogoče doseči [3]. Z neposrednimi izkušnjami odkrivajo nove informacije ter nadgrajujejo oziroma povezujejo predhodno znanje s konkretnimi primeri iz narave. Raziskave kažejo, da ima terensko delo pozitiven vpliv na razumevanje učne snovi, motivacijo in odnos do narave [4]. Učenci pri tem razvijajo tudi socialne spretnosti, sodelovanje in odgovoren odnos do okolja [1].

Metodo učenja v naravi ali terensko delo različni avtorji opredeljujejo z različnih vidikov. Scott in sod. [5] ga opisujejo kot aktivnosti, namenjene spoznavanju narave zunaj učilnice, ki lahko trajajo od nekaj minut do večdnevni aktivnosti. Učenje v naravi omogoča razvoj različnih kompetenc, kot so opazovanje, reševanje problemov in kritično mišljenje, hkrati pa spodbuja zanimanje za biologijo [1]. Kljub temu se učitelji zaradi organizacijskih omejitev, pomanjkanja samozavesti in časovnih omejitev redkeje odločajo za njegovo uporabo [6].

Pomemben vidik sodobnega poučevanja pomeni tudi medpredmetno povezovanje, ki omogoča celostno obravnavo znanja. V hitro spreminjajoči se družbi je za razumevanje kompleksnih pojavov treba povezovati različne znanstvene discipline [7]. Interdisciplinarni pristop spodbuja razvoj kritičnega mišljenja, sklepanja in ustvarjalnosti ter omogoča boljše razumevanje realnega sveta [8].

Ker ugotavljamo pozitiven vpliv učenja v naravi in medpredmetnega povezovanja, smo v okviru pouka naravoslovja oblikovali učno aktivnost »ura naravoslovja v naravi« za učence 6. razreda osnovne šole. Aktivnost temelji na metodi terenskega dela in vključuje medpredmetno povezovanje bioloških, kemijskih in fizikalnih vsebin.

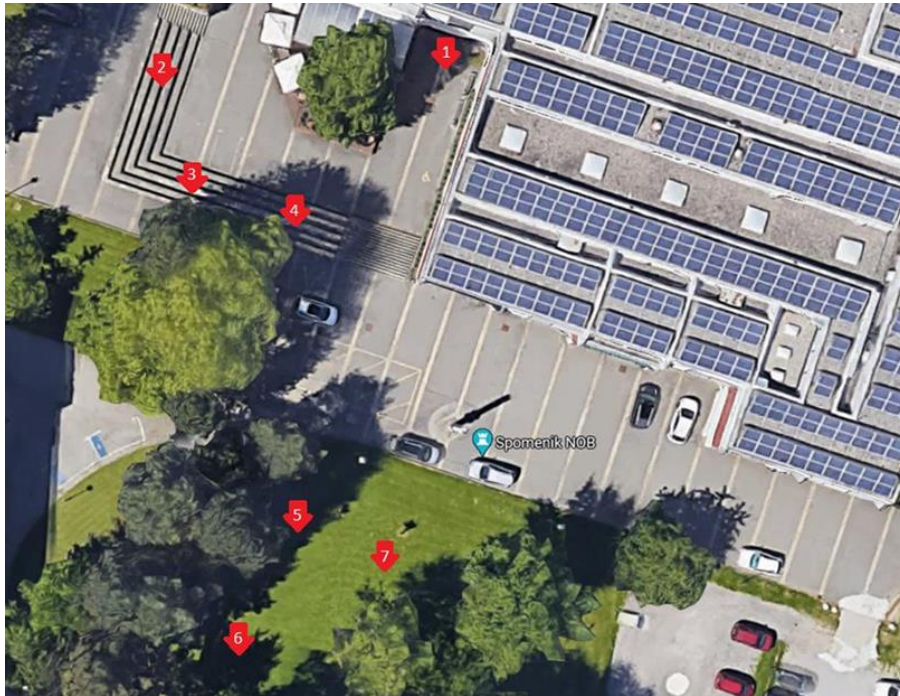
**Namen prispevka** je predstaviti zasnovo, izvedbo in evalvacijo učne ure v naravi kot didaktičnega pristopa za poučevanje naravoslovnih vsebin.

**Cilji prispevka** so:

- oblikovati nabor delovnih postaj za izvedbo učne ure v naravi,
- prikazati možnosti medpredmetnega povezovanja,
- ovrednotiti izvedbo aktivnosti na podlagi opažanj in odzivov učencev.

## **2 MATERIAL IN METODE DELA**

Pred izvedbo ure naravoslovja v naravi smo pregledali učni načrt za 6. razred, izbrali vsebine in oblikovali aktivnosti. Pripravili smo delovni material, ki je vključeval delovne liste, navodila in zemljevid z označenimi delovnimi postajami (slika 1). Delovni listi, uporabljeni pri posameznih delovnih postajah, so prikazani v prilogah (priloge 1–7). Pripravili smo tudi potrebne pripomočke za izvedbo delavnice. Pred pričetkom ure smo učencem razložili potek delavnice in jim podali natančna navodila za delo.



Slika 1: Zemljevid delovne površine z oštevilčenimi delovnimi postajami (Google Maps).

## 2.1 Material

Pri izvedbi ure naravoslovja v naravi smo uporabili veliko pripomočkov. Vsak učenec je imel svoje pisalo in je na delovni postaji dobil ustrezen delovni list. Za lažjo predstavitev uporabljenih pripomočkov jih bomo združili v skupine glede na delovno postajo, kjer so potrebni.

Za delovno postajo *Kaj vpliva na rast mahu?* smo ob drevesih potrebovali:

- digitalni merilec temperature,
- merilec vlage,
- merilec za jakost svetlobe,
- pripravljena navodila za delo s pripomočki.

Za delovno postajo *Katera cvetlica je to?* smo ob travniku s cveticami potrebovali:

- določevalni ključ *Katera cvetlica je to?* [9].

Za delovno postajo *Katero drevo je to?* smo ob izbranih drevesih potrebovali:

- določevalni ključ *Katero drevo je to?* [10],
- papirnate oblike različnih barv,
- pripravljen zemljevid izbranih dreves,
- pripravljen enostaven določevalni ključ.

Za delovno postajo *Kako določimo starost drevesa?* smo ob izbranih drevesih potrebovali:

- prerez debla z vidnimi letnicami,
- kalkulator,
- krojaški meter.

Za delovno postajo *Mozaik listne ploskve* smo potrebovali:

- mikroskop,
- pribor za mikroskopiranje,
- rastlino,
- prozoren lak za nohte,
- objektno in krovno steklo,
- kapalko z vodo.

Za delovno postajo *Semena in plodovi* smo potrebovali:

- semena (arašidi, orehova jedrca, fižol, grah, seme nektarine, seme jabolka, bučno seme in sončnično seme),
- papir temne barve,
- nož,
- jodovica,
- kapalka,
- krožnik,
- pripravljena navodila s potekom dela poskusov.

Za delovno postajo *Ali olje res plava na vodi?* smo potrebovali:

- vodo,
- olje,
- led,
- kozarec za vlaganje,
- pomarančo,
- tehtnico,
- merilni valj,
- vnaprej pripravljen tekočinski stolpec.

## 2.2 Organizacija delovnih postaj

Učenci so oblikovali skupine po tri, pri čemer so se premikali med posameznimi delovnimi postajami in reševali naloge. Navodila za izvedbo so bila podana tudi na delovnih listih in dodatnih gradivih. Za izvedbo nalog na posamezni delovni postaji so imeli na voljo 15 minut, nato so prešli na naslednjo postajo. Vsaka skupina je na začetku prejela delovne liste, številko skupine ter številko prve delavnice. Ob tem smo jih seznanili z zemljevidom delovnih postaj (slika 1) in navodili na delovnih listih. Delovni listi za drugo in tretjo delovno postajo so bili nameščeni na mestu delovnih postaj v mapah, označenih s številkami skupin. Delo je potekalo samostojno. V primeru vprašanj ali težav z izvedbo aktivnosti je bila v bližini prisotna izvajalka, ki je učence usmerila v nadaljevanje izvedbe.

Pred začetkom aktivnosti smo učencem razložili sistem namigov za prehod med delovnimi postajami. Ob tem smo poudarili pomen skupinskega dela, primerne razdelitve dela znotraj skupine ter spoštljivega odnosa do sošolcev in narave.

Ob pripravi didaktično-metodološke aktivnosti smo upoštevali aktualne trende v poučevanju bioloških vsebin. Izhajali smo iz metode terenskega dela, zato smo dejavnosti izvedli v naravi. Vključiti smo želeli medpredmetno povezovanje ter metodo didaktične igre. Navdih smo črpali iz aktualnih "sob pobega" oziroma "lova na zaklad", kjer udeleženci z reševanjem nalog dobijo namige za nadaljevanje aktivnosti. Ideja je bila, da bi učenci na vsaki delovni postaji z reševanjem nalog pridobili znanje, na podlagi katerega bi prepoznali namig za naslednjo

postajo. Šele takrat lahko nadaljujejo z delom na naslednji točki. Opisana dinamika izvedbe delavnic zahteva pripravo več postaj, kot je skupin učencev. V našem primeru smo imeli pet skupin po tri učence, zato smo pripravili sedem delovnih postaj. Tako smo zagotovili aktivno sodelovanje vseh učencev ves čas trajanja aktivnosti ter oblikovali vsebinsko smiselne kombinacije delovnih postaj za vse skupine.

Za namen naravoslovne ure v naravi smo oblikovali naslednje delovne postaje:

1. *Kaj vpliva na rast mahu?*; delovni list je predstavljen v prilogi 1.
2. *Katera cvetlica je to?*; delovni list je predstavljen v prilogi 2.
3. *Katero drevo je to?*; delovni list je predstavljen v prilogi 3.
4. *Kako določimo starost drevesa?*; delovni list je predstavljen v prilogi 4.
5. *Mozaik listne ploskve*; delovni list je predstavljen v prilogi 5.
6. *Semena in plodovi*; delovni list je predstavljen v prilogi 6.
7. *Ali olje res plava na vodi?*; delovni list je predstavljen v prilogi 7.

### 2.3 Vključevanje v učni načrt

Delavnica je bila izvedena junija 2024, ko so učenci že usvojili večino vsebin učnega načrta [11]. Po posvetu z matičnim učiteljem smo izpustili vsebine iz sklopa *Vpliv človeka na okolje* [11], saj tega sklopa učenci še niso v celoti obravnavali. Učitelj je predlagal, naj se aktivnosti osredotočajo na tematski sklop *Živa narava* [11].

Od sedmih oblikovanih delovnih postaj jih je bilo šest zasnovanih na bioloških vsebinah in so se nanašale na vsebinski sklop *Živa narava* [11]. Ena delovna postaja je bila kemijsko in fizikalno obarvana in se je navezovala na vsebinski sklop *Snovi* [11]. Za vsako delovno postajo smo pripravili delovni list z opisom dela, nalogami in prostorom za zapis rezultatov ter namigom za naslednjo postajo, kar je vključevalo tudi orientacijo učencev.

### 2.4 Operativni cilji delavnice

Posamezne delovne postaje smo oblikovali v okviru zastavljenih operativnih ciljev. Te smo razdelili v tri skupine: a) kognitivne, b) konativne in c) psihomotorične, zapisane so v nadaljevanju.

Učenci na *kognitivni ravni* lahko dosežejo naslednje cilje:

- primerjajo izmerjene vrednosti temperature zraka in tal, vlage tal ter jakosti svetlobe za drevo brez mahu in drevo z mahom (analiza);
- iz izmerjenih vrednosti prejšnje linije predvidijo, v kakšnem okolju mahovi bolje uspevajo (analiza);
- s pomočjo slikovnih določevalnih ključev določijo travniško rastlinsko vrsto (uporaba);
- na podlagi besedilnega opisa določijo travniško rastlinsko vrsto (uporaba);
- s pomočjo pripravljenega opisnega določevalnega ključa določijo drevesno vrsto (uporaba);
- iz besednega opisa rastline sklepajo na opisano drevesno vrsto (analiza);
- na osnovi metode štetja letnic ter merjenja obsega debla določijo starost drevesnega primerka (uporaba);
- za določanje starosti dreves ovrednotijo natančnost uporabljenih metod (štetje letnic ter merjenje obsega), (vrednotenje);
- poimenujejo sestavne dele mikroskopa (razumevanje);

- pripravijo mikroskopski preparat za opazovanje listnih rež (uporaba);
- na mikroskopskem preparatu prepoznajo listne reže (razumevanje);
- poimenujejo pripravljena semena rastlin (pomnjenje);
- na podlagi kriterijev raznašanja semen s pomočjo vetra, vode, živali ali s pomočjo ljudi razvrstijo plodove in semena v smiselne skupine (razumevanje);
- iz rezultatov poskusa sklepajo na prisotnost olj v semenih in plodovih (analiza);
- iz rezultatov poskusa sklepajo na prisotnost škroba v semenih in plodovih (analiza);
- predvidijo položaj kocke ledu v različnih tekočinah (uporaba);
- izračunajo gostoto pomaranče (uporaba);
- glede na različne gostote tekočin predvidijo zaporedje plasti v tekočinskem stolpcu (analiza).

Učenci pri *konativnih ciljih*:

- razvijajo sposobnosti medsebojnega komuniciranja;
- razvijajo sposobnosti dela v skupini;
- razvijajo sposobnosti logičnega mišljenja in sklepanja;
- razvijajo pozitiven in odgovoren odnos do narave in živih bitij;
- spoštujejo različna mnenja;
- sklepajo iz že znanega.

Učenci pri *psihomotornih ciljih*:

- aktivno sodelujejo;
- sledijo navodilom;
- razvijajo sposobnosti opazovanja;
- razvijajo sposobnosti mikroskopiranja;
- razvijajo sposobnosti učinkovite uporabe zemljevida;
- urijo se v rabi merilnih pripomočkov;
- urijo se v izvedbi poskusov;
- urijo se v pripravi lastnih mikroskopskih preparatov;
- urijo se v orientaciji.

## 2.5 Vsebina delovnih postaj

V nadaljevanju so podrobneje predstavljene izbrane delovne postaje, medtem ko so delovni listi za vse postaje vključeni v prilogah (priloge 1–7).

### 1. Delovna postaja: Kaj vpliva na rast mahu?

V parku pred Fakulteto za naravoslovje in matematiko Univerze v Mariboru lahko najdemo različne predstavnike rastlin, tudi mahove, zato smo jih vsebinsko želeli vključiti in njihovo nahajališče povezati z vplivom abiotskih dejavnikov okolja na rast in razvoj. Iz slednjega razloga so učenci merili izbrane abiotske dejavnike (temperaturo, količino vlage in svetlobe). Ob meritvah so se urili v rokovanju z njim neznanimi merilnimi pripomočki (digitalni merilec temperature, merilec vlage v tleh ter merilec jakosti svetlobe), v pridobivanju in zbiranju podatkov ter na osnovi pridobljenih rezultatov v oblikovanju zaključkov o rasti mahu. Naloga učencev je bila izmeriti izbrane abiotske dejavnike okolja ob drevesu, na katerem raste mah, in ob drevesu brez mahu. V ta namen so na delovni postaji bili merilni pripomočki in plastificirana navodila za uporabo (slika 2). Navodila smo plastificirali zaradi zaščite in trajnejše uporabe za

vse skupine. Iz dobljenih podatkov so učenci morali predvideti ustrezne pogoje za rast mahu. Delovni list za to dejavnost je predstavljen v prilogi 1.



Slika 2: Pripomočki delovne postaje *Kaj vpliva na rast mahu?* (od leve proti desni: navodila, digitalni merilnik vlage v tleh, merilnik jakosti svetlobe, digitalni merilnik temperature).

## 2. Delovna postaja: *Katero drevo je to?*

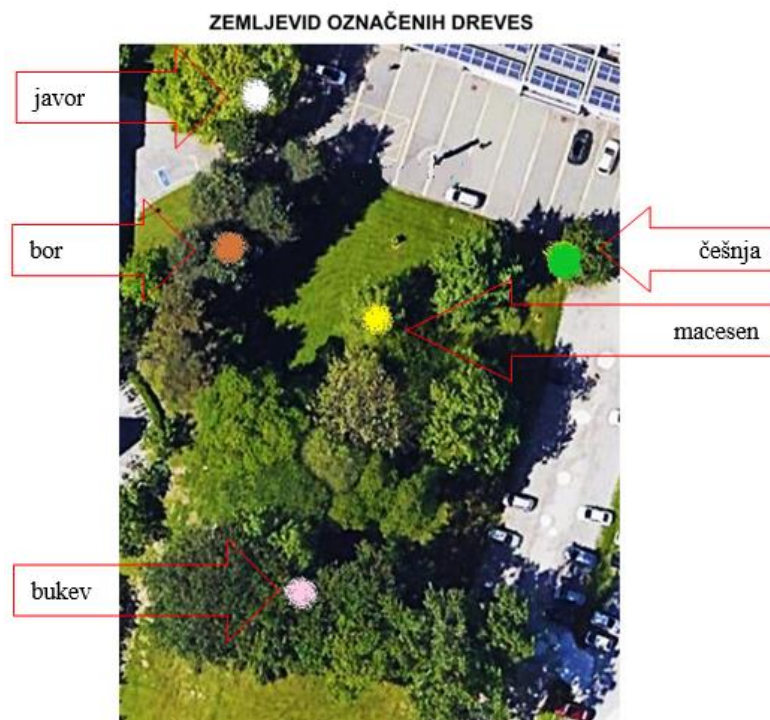
Učence je na 2. delovni postaji pričakal pripravljen enostaven določevalni ključ in zemljevid označenih dreves, ki so jih med aktivnostjo obravnavali (slika 3). Delovno gradivo je bilo plastificirano, saj smo tako zagotovili dolgotrajnejšo in večkratno uporabo. Učenci so se premikali od drevesa do drevesa in jih s pomočjo določevalnega ključa določili. Del rastline, po katerem so prepoznali drevesno vrsto, so morali skicirati na delovni list (priloga 3).



Slika 3: Učenci v izvajanju delovne postaje *Katero drevo je to?* (navodila in določevalni ključ).

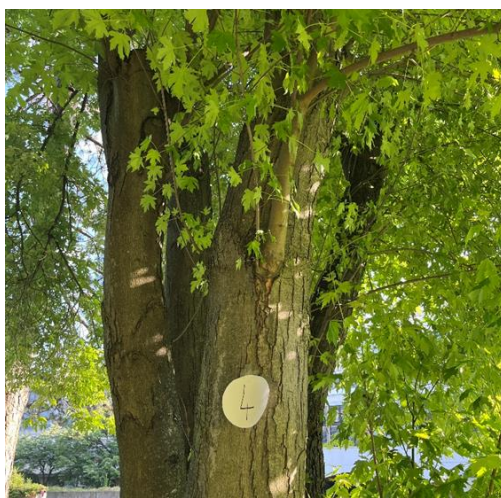
Za delovno postajo *Katero drevo je to?* smo izbrali točno določena drevesa, ki naj bi jih učenci prepoznali. V ta namen smo oblikovali določevalni ključ drevesnih vrst ter zemljevid z označenimi drevesi, ki so jih morali poimenovati. Prvi korak oblikovanja delovne postaje je bil ogled dreves v parku pred Fakulteto za naravoslovje in matematiko Univerze v Mariboru. Na

podlagi tega smo izbrali primerna drevesa za določevanje. Po izboru dreves smo s pomočjo aplikacije Google Maps [12] zajeli posnetek parka iz ptičje perspektive in ga prenesli v program Slikar. Drevesa smo označili z barvnimi označbami, saj ta pristop učencem omogoča poljubno izbiro zaporedja določevanja (slika 4).



Slika 4: Učenci v izvajanju delovne postaje *Katero drevo je to?* (Google Maps).

Drevesa smo označili z barvnimi krogi. Pri izbiri barve v programu Slikar smo zato morali biti pazljivi, da smo imeli ustrezni barvni papir, s katerim smo označili drevo (slika 5). Številka, ki je na barvnem krogu, je povezana z nalogo in namigom za naslednjo delovno postajo.



Slika 5: Označeno drevo za delovno postajo *Katero drevo je to?*

Oblikovali smo enostaven določevalni ključ, kjer so se učenci morali najprej odločiti, ali je drevo listavec ali iglavec. Na osnovi primerjanja kratkih opisov in slik je sledilo določevanje drevesne vrste.

### 3. Delovna postaja: Semena in plodovi

Vsebina delavnice je preplet kemije in biologije s poudarkom na organskih snoveh. Oblikovali smo delovno postajo, kjer so učenci raziskovali prisotnost olj in škroba v različnih plodovih in semenih ter se pogovorili o njihovem pomenu. V drugem delu postaje so učenci semena in plodove razvrščali glede na način raznašanja.

Navodila za izvedbo poskusov smo pripravili kot dodatno gradivo, ki je učence pričakalo na delovni postaji. Z zavedanjem, da je delovna postaja časovno krajša od ostalih, smo gradivo plastificirali in izrezati v obliko sestavljanke (slika 6). Prva naloga na tej delovni postaji je bila sestaviti sestavljanke in tako razkriti navodila za izvedbo poskusov



Slika 6: Učenci pri sestavljanju navodil za delavnico *Semena in plodovi*.

Učenci so izvedli dva poskusa z različnimi semeni. Na razpolago so imeli: arašide, orehovo jedrce, fižol, grah, seme nektarine, seme jabolka, bučno seme in sončnično seme. V prvem poskusu so določali prisotnost škroba. To so naredili s prerezom semena s skalpelom in nanje kapljali jodovico. Zaradi uporabe kemikalije so učenci dobili dodatna varnostna opozorila. Prav tako je bila ves čas izvedbe poskusa prisotna odrasla oseba. Na podlagi obarvanja so učenci predvidevali prisotnost škroba in svojo odločitev zabeležili na delovni list. V drugem poskusu so v semenih določali prisotnost olj, in sicer tako, da so prerezana semena podrgnili po papirju temne barve in opazovali morebitni mastni madež (slika 7). Ugotovitve so zabeležili na delovni list. Nato so se pogovorili in semena ter plodove razvrstili po načinu raznašanja (s pomočjo vode, vetra, živali ali ljudi). Delovni list za to dejavnost je predstavljen v prilogi 6.

Slika 7: Rezultat delovne postaje *Semena in plodovi*.

## 2.6 Namigi za prehod na naslednjo delovno postajo

Za izvedbo ure naravoslovja v naravi so učenci imeli na voljo 60 minut. V tem času je vsaka skupina učencev lahko izvedla tri delovne postaje.

Oblikovali smo namige za prehode na naslednjo delovno postajo. Le-te smo v obliki naloge dodali na delovne liste, ki se je nanašala na usvojeno znanje delovne postaje. Naloge z namigi smo oblikovali za vsako delovno postajo. Ker skupine niso vedno prehajale na isto postajo, smo namige preoblikovali za vsako skupino posebej. Izjema so bile zadnje postaje, kjer ni bilo več prehoda. Zapis zaporedja delavnic ter nalog namigov je prikazan v tabeli 1.

Število skupine	Kombinacije delavnic (število delavnice ter ime)	Namigi za naslednjo delavnico
1	(7) <i>Katero drevo je to?</i>  (6) <i>Kaj vpliva na rast mahu?</i>  (2) <i>Kako določiti starost drevesa?</i>	Učenci so iz opisa prepoznali drevesno vrsto, na kateri je bila barvna oznaka s številko naslednje delovne postaje (6). Učenci so poiskali embalažo merilca za vlago tal, na kateri je bila številka naslednje postaje (2).
2	(2) <i>Kako določiti starost drevesa?</i>  (5) <i>Katera cvetlica je to?</i>  (1) <i>Mozaik listne ploskve</i>	Učenci so poiskali štor, star 41 let. V njegovi bližini je bila številka naslednje postaje (5). Učenci so iz opisa rastlinske vrste prepoznali cvetlico. V bližini cvetlice je bila številka naslednje delovne postaje (1).
3	(1) <i>Mozaik listne ploskve</i>  (3) <i>Ali res olje plava na vodi?</i>	Deli mikroskopa so bili označeni s številkami. Učenci so prepoznali objektiv, na katerem je bila številka naslednje postaje (3).

	(4) <i>Semena in plodovi</i>	Učenci so ugotovili, katera zapovrstjo je bila voda v tekočinskem stolpcu, kar jim je dalo številko naslednje delovne postaje (4).
--	------------------------------	--

Tabela 1: Kombinacije delavnic za skupine ter namigi za prehod na naslednjo postajo.

### 3 REZULTATI IN DISKUSIJA

Z aktivnostjo ura naravoslovja v naravi smo želeli učencem 6. razreda osnovne šole ustvariti poučno, a hkrati zabavno učno izkušnjo zunaj razreda.

Analiza izvedbe delavnice in pregled izpolnjenih delovnih listov sta pokazala, da so učenci večino nalog uspešno rešili in razumeli navodila za delo. Največ težav se je pojavilo pri nalogah, ki so zahtevale uporabo zemljevida ter določanje rastlinskih vrst, kar nakazuje na potrebo po dodatnem razvijanju orientacijskih spretnosti in uporabe določevalnih ključev. Učenci so bili pri večini aktivnostih samostojni, pri zahtevnejših nalogah (npr. mikroskopiranje in izvedba poskusov) pa so potrebovali dodatno usmerjanje izvajalcev. Rezultati kažejo, da je takšen način dela primeren za utrjevanje znanja in razvoj praktičnih spretnosti, hkrati pa omogoča vpogled v težavnost posameznih vsebin. Ugotovitve so skladne z ugotovitvami drugih raziskav, ki poudarjajo, da učenje v naravi prispeva k boljšemu razumevanju vsebin in večji motivaciji učencev [4, 13].

V okviru delavnice smo pripravili nabor delovnih postaj, ki smo jih smiselno povezali z namigi, uporabo zemljevidov in usmerjanjem pozornosti na natančno opazovanje. Cilj je bil motivirati učence za učenje naravoslovnih vsebin, življenjsko povezovanje naravoslovnih vsebin in dvig ravni zanimanja za naravoslovne predmete. Predstavljeno aktivnost smo časovno omejili na 60 minut. Takšno delavnico lahko prilagodimo različnim časovnim razponom. Čeprav smo aktivnost izvedli v urbanem okolju in ne v neokrnjeni naravi, so učenci svoje znanje o rastlinah nadgradili z neposrednim stikom z realnimi primeri. Rezultati kažejo, da smo z dejavnostjo dosegli zastavljene cilje in namen ter pripravili gradivo za zanimivo in poučno aktivnost v naravi, ki se lahko prilagodi potrebam učiteljev.

Učencem so bila splošna navodila ter navodila na delovnih postajah razumljiva. Samostojno so izvajali delavnico, kot smo si zamislili. Največ težav se je pojavljalo pri branju zemljevida. Menimo, da bi morali na zemljevid dodati smeri neba ter z učenci pred začetkom dela ponoviti smeri neba.

Kot smo pričakovali, so učenci potrebovali nekoliko več pomoči na delovnih postajah *Semena in plodovi*, *Ali olje res plava na vodi?* ter *Mozaik listne ploskve*. Zato sta dve od treh izvajalk pomagali tem skupinam pri izvedbi. Tretja izvajalka je tako pomagala učencem na delovnih postajah v gozdu oziroma na travniku. Na delovni postaji *Mozaik listne ploskve* je bila potrebna prisotnost izvajalke, saj nismo dodali navodil za pripravo mikroskopskega preparata. Le-te so učenci dobili ustno.

Pri delovnih postajah *Semena in plodovi*, *Ali olje res plava na vodi?* ter *Kako določimo starost drevesa?* smo ugotovili, da se nahajajo preblizu. Učenci so namreč slišali odgovore prejšnjih skupin in kasneje niso izvedli delovne postaje v celoti. Te delovne postaje bi bilo smiselno razporediti prostorsko bolj narazen.

Opazili smo tudi, da so učenci imeli težave z določanjem cvetočih rastlin na travniku. Tudi sami so kasneje te težave izrazili in povedali, da bi raje to dejavnost zamenjali. Nadgradnja delovne postaje bi bila lahko primerjanje uporabe knjižnih določevalnih ključev ter spletnih oziroma mobilnih aplikacij za določevanje. Na ta način bi lahko povezali znanje naravoslovja, kritično mišljenje in vedno pomembnejše digitalne veščine.

Drugih težav pri izvajanju aktivnosti nismo zaznali. Učenci so se držali navodil in navodila posameznih delovnih postaj tudi razumeli. Delali so samostojno z ustrezno medsebojno

komunikacijo in spoštovanjem. Opaziti je bilo, da so jih dodatno pritegnili namigi za naslednjo postajo. Pri tej nalogi je bilo mogoče opaziti največ zanimanja in vneme s strani učencev. Vendar so tudi preostale naloge na delovnih listih bile v večini pravilno rešene.

Po koncu aktivnosti smo bile izvajalke zadovoljne. Potekala je v večini po naših načrtih in pričakovanjih. Učenci so tudi v pogovoru o opravljenem delu izkazali navdušenost nad tem tipom poučevanja. Delovne postaje bomo v prihodnosti še uporabile pri svojem delu.

### 3.1 Vtisi učencev

Po koncu dejavnosti so učenci izkazali navdušenje nad pripravljeno delavnico in izvedbo učne ure. Navdušenje učencev nad izvedbo ure v naravi so v raziskavi zasledili tudi Marchant in sod. [13], saj so ugotovili, da učenje v naravi v učencih spodbudi občutke svobode ter zabave. Eden izmed učencev je izjavil, da se pri učenju v učilnici počuti pod stresom, medtem ko ga učenje v naravi osrečuje. Veliko udeležencev naše delavnice je izrazilo veselje nad učenjem v naravi. Večina učencev je bila mnenja, da bi moralo biti več takšnih dejavnosti in da bi se jih z veseljem ponovno udeležili. Sjöblom in Svens [14] sta ugotovila podobno, da imajo učeči raje učenje v naravi kot učenje v učilnici. Učenci v tej raziskavi so predvsem poudarili različne učne dejavnosti, ki jih v učilnici ni mogoče izvesti. Primer tega je bilo opazovanje rastlin ter njihovo vključevanje v različne aktivnosti. Ker želimo dejavnosti izboljšati, smo učence prosili za predloge morebitnih sprememb, izboljšav, dopolnitev. Večina jih je predlagala več časa za izvajanje vseh aktivnosti. Nekateri so izrazili željo po menjavi dejavnosti določevanja cvetlic, saj jim je bila ta delovna postaja (pre)zahtevna.

## 4 ZAKLJUČEK

Menimo, da je metoda terenskega dela za poučevanje predvsem bioloških vsebin zelo pomembna, saj konkretizira predhodno pridobljeno znanje z metodami branja in poslušanja. Z odhodom v naravo lahko izvedemo različne aktivnosti, s katerimi bogato nadgradimo poučevanje v učilnici, znanje otrok pa bo trajnejše. Pri izvajanju dejavnosti v naravi lahko učenci uporabljajo večino čutil, kar poveča trajnost pridobljenega znanja. Na to vpliva tudi možnost ustvarjanja unikatnih in ustvarjalnih izkušenj, o katerih učenci dlje časa razmišljajo [15]. Na podlagi izvedbe lahko zaključimo, da takšna oblika poučevanja omogoča učinkovito povezovanje teorije s prakso ter spodbuja aktivno učenje učencev. Prispevek ponuja konkretne smernice za učitelje, ki želijo vključiti krajše oblike terenskega dela v redni pouk.

## Literatura

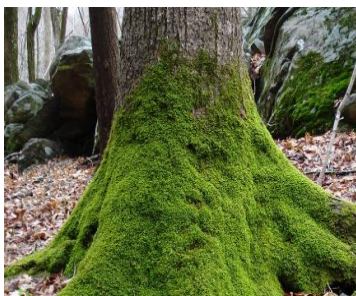
- [1] Arianti, Y., Aminatun, T. (2019, June). An analysis of outdoor learning towards students' outcomes in learning biology. In *Journal of Physics: Conference Series*, 1241(1), 012061. IOP Publishing. DOI: 10.1088/1742-6596/1241/1/012061
- [2] Fleischner, T. L., Espinoza, R. E., Gerrish, G. A., Greene, H. W., Kimmerer, R. W., Lacey, E. A., ..., Zander, L. (2017). Teaching biology in the field: importance, challenges, and solutions. *BioScience*, 67(6), 558–567. DOI: 10.1093/biosci/bix036
- [3] Ezechi, N. G. (2018). Influence of field trip in teaching and learning of biology. *International Journal of Engineering & Scientific Research*, 6(10), 32–37. Dostopno na: [https://www.academia.edu/43389463/INFLUENCE\\_OF\\_FIELD\\_TRIP\\_IN\\_TEACHING\\_AND\\_LEARNING\\_OF\\_BIOLOGY](https://www.academia.edu/43389463/INFLUENCE_OF_FIELD_TRIP_IN_TEACHING_AND_LEARNING_OF_BIOLOGY).

- [4] Glackin, M., Greer, K. (2021). Making the case for A-level biology residential fieldwork: what has nature got to do with it? *School Science Review*, 102(381), 21–26. Dostopno na: <https://kclpure.kcl.ac.uk/portal/en/publications/making-the-case-for-a-level-biology-residential-fieldwork-what-ha>.
- [5] Scott, G. W., Boyd, M., Scott, L., Colquhoun, D. (2015). Barriers to biological fieldwork: What really prevents teaching out of doors? *Journal of Biological Education*, 49(2), 165–178. DOI: 10.1080/00219266.2014.914556
- [6] Kervinen, A., Uitto, A., Juuti, K. (2020). How fieldwork-oriented biology teachers establish formal outdoor education practices. *Journal of Biological Education*, 54(2), 115–128. DOI: 10.1080/00219266.2018.1546762
- [7] Oudenampsen, J., Van De Pol, M., Blijlevens, N., Das, E. (2023). Interdisciplinary education affects student learning: a focus group study. *BMC medical education*, 23(169). DOI: 10.1186/s12909-023-04103-9
- [8] You, H. S. (2017). Why teach science with an interdisciplinary approach: history, trends, and conceptual frameworks. *Journal of Education and Learning*, 6(4), 66–77. DOI: 10.5539/jel.v6n4p66
- [9] Spohn, M., Spohn, R. (2008). *Katera cvetlica je to?* Olševsek: Narava.
- [10] Spohn, M., Spohn, R. (2008). *Katero drevo je to?* Olševsek: Narava.
- [11] Vihar B., Zupančič G., Gilčvert Berdnik D., Vičar M. (2011). *Učni načrt. Program osnovna šola. Naravoslovje*. Ljubljana: Ministrstvo RS za šolstvo in šport: Zavod RS za šolstvo. Pridobljeno: 23. 5. 2023: <https://www.gov.si teme/programi-in-ucni-naacrti-v-osnovni-soli/>.
- [12] Google Maps. Fakulteta za naravoslovje in matematiko, Koroška cesta 160, 2000 Maribor. Pridobljeno: 26. 5. 2023: [https://www.google.com/maps/place/Fakulteta+za+naravoslovje+in+matematiko+Univerze+v+Mariboru/@46.5631332,15.6236905,118m/data=!3m1!1e3!4m6!3m5!1s0x476f77c96c7f31f7:0xabfce737bc639184!8m2!3d46.5636726!4d15.6242866!16s%2Fg%2F121cb2s7?entry=tту&g\\_ep=EgoyMDI1MDQyMy4wIKXMDS0ASAFQAw%3D%3D](https://www.google.com/maps/place/Fakulteta+za+naravoslovje+in+matematiko+Univerze+v+Mariboru/@46.5631332,15.6236905,118m/data=!3m1!1e3!4m6!3m5!1s0x476f77c96c7f31f7:0xabfce737bc639184!8m2!3d46.5636726!4d15.6242866!16s%2Fg%2F121cb2s7?entry=tту&g_ep=EgoyMDI1MDQyMy4wIKXMDS0ASAFQAw%3D%3D).
- [13] Marchant, E., Todd, C., Cooksey, R., Dredge, S., Jones, H., Reynolds, D., ..., Brophy, S. (2019). Curriculum-based outdoor learning for children aged 9-11: A qualitative analysis of pupils' and teachers' views. *PloS one*, 14(5), e0212242. DOI: 10.1371/journal.pone.0212242
- [14] Sjöblom, P., Svens, M. (2019). Learning in the Finnish outdoor classroom: Pupils' views. *Journal of Adventure Education and Outdoor Learning*, 19(4), 301–314. DOI: 10.1080/14729679.2018.1531042
- [15] Taş, E., Gülen, S. (2019). Analysis of the influence of outdoor education activities on seventh grade students. *Participatory educational research*, 6(2), 122–143. DOI: 10.17275/per.19.17.6.2

## Priloge

### Priloga 1: Delovni list za postajo *Kaj vpliva na rast mahu?*

## Kaj vpliva na rast mahu?



Si se kdaj sprehajal po gozdu in spraševal, zakaj se na nekaterih drevesih pojavlja mah. Kaj vpliva na rast na določenih drevesih in ne vseh? Z znanjem o rastlinah, ki si ga pridobil v 6. razredu in današnjo vajo, boš ugotovil, kaj vpliva na rast mahov.

Med nežive oziroma **abiotske dejavnike** uvrščamo temperaturo, vlago, jakost svetlobe, padavine itd. Merimo jih z merilci. Temperatura zraka, ki jo izmerimo v Mariboru, ne bo enaka temperaturi zraka izmerjeni v Kopru ali celo v Maroku. S primerjanjem vrednosti abiotskih dejavnikov, vidimo kako rastline reagirajo na nje.

**MATERIAL:** Merilci temperature, vlage, jakosti svetlobe, drevesi.

**NALOGA:** Primerjati izmerjene vrednosti abiotskih dejavnikov (temperatura zraka in tal, vlaga tal ter jakost svetlobe) pod dvema drevesoma in ugotoviti boljše razmere za mah. Rezultat vpiši v preglednici 1 in 2.

### **REZULTATI:**

*Preglednica 1: Vrednosti meritev za drevo brez mahu.*

Preiskovan dejavnik	Meritev
temperatura tal [°C]	
temperatura zraka [°C]	
vlaga tal [%]	
jakost svetlobe [lux]	

*Preglednica 2: Vrednosti meritev za drevo z mahom.*

Preiskovan dejavnik	Meritev
temperatura tal [°C]	
temperatura zraka [°C]	
vlaga tal [%]	
jakost svetlobe [lux]	

### **VPRAŠANJA:**

- V skupini se pogovorite in primerjajte dobljene vrednosti. Ob zapisanih trditvah obkrožite drevo, katero pravilno nadaljuje trditev.

Višja temperatura zraka je v okolici	drevesa brez mahu.	drevesa z mahom.
Višja temperatura tal je v okolici	drevesa brez mahu.	drevesa z mahom.
Večji odstotek vlage tal je v okolici	drevesa brez mahu.	drevesa z mahom.
Večja jakost svetlobe je v okolici	drevesa brez mahu.	drevesa z mahom.

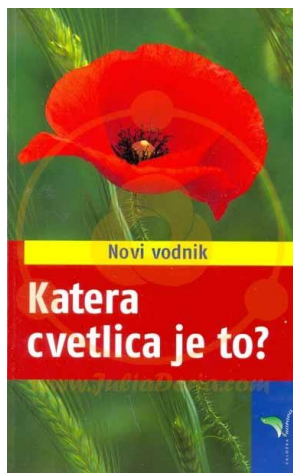
- Glede na rezultate navedi okolje, kje mahovi bolje uspevajo.

**NADALJEVANJE:** S pomočjo znanja, pridobljenega na delovni postaji, pridobi namig za naslednjo delovno postajo.

Med materialom, ki je bil potreben za izvedbo vaje, poišči škatlo higrometra in našel boš namig (številko) naslednje postaje.

**Priloga 2:** Delovni list za postajo *Katera cvetlica je to?*

## Katera cvetlica je to?



Prepoznaš cvetlice? Po katerih znakih ločujemo rastline?

Raznoliki barvni cvetovi, zanimive oblike listov, vse to in še več boš danes opazoval.

Za poimenovanje rastlin so znanstveniki določili sistem za razlikovanje med njimi in njihovo razvrščanje. Rastline razvrščamo s pomočjo **določevalnih ključev**. Obstaja več različnih določevalnih ključev, kot so: slikovni, mrežni, tekstovni, vsem pa je skupno, da se na vsakem koraku odločamo med dvema lastnostima organizma, dokler ne pridemo do končne določitve vrste.

**MATERIAL:** Določevalni ključi rastlin, rastline.

**NALOGA:** S pomočjo določevalnega ključa poimenovati rastline. Od vsakega predstavnika v preglednico 1 skicirajte del rastline, ki je pomemben za prepoznavanje.

### **REZULTATI:**

*Preglednica 1: Poimenovanje rastlin in skicirani rastlinski deli.*

Rastlinska vrsta	Rastlinski del (list, cvet)
slovensko ime: _____	
slovensko ime: _____	
slovensko ime: _____	
slovensko ime: _____	

**NADALJEVANJE:** S pomočjo znanja, pridobljenega na delovni postaji, pridobi namig za naslednjo delovno postajo.

S pomočjo knjige poišči rastlino, ki ima več belih cvetov v košku. Cveti od januarja do novembra. Po hladnih nočeh se cvetovi obarvajo rahlo rdečkasto. Znanstveniki jo poznajo pod imenom *Bellis perennis*.

Ko prepoznaš opisano rastlino, jo poišči na travniku in razkrija se ti bo številka naslednje postaje.

### Priloga 3: Delovni list za postajo *Katero drevo je to?*

## Katero drevo je to?



Prepoznaš drevesa? Po katerih znakih se med seboj razlikujejo predstavniki listavcev in iglavcev? Tokrat boste imeli možnost določiti nekaj drevesnih vrst.

Za poimenovanje rastlin so znanstveniki določili sistem za razlikovanje med njimi in njihovo razvrščanje. Rastline razvrščamo s pomočjo **določevalnih ključev**. Obstaja več različnih določevalnih ključev, kot so: slikovni, mrežni, tekstovni, vsem pa je skupno, da se na vsakem koraku odločamo med dvema lastnostima organizma, dokler ne pridemo do končne določitve vrste.

**MATERIAL:** Določevalni ključ, priloga A in priloga B.

**NALOGA:** S pomočjo priloženega določevalnega ključa določite drevesne vrste. Od vsakega predstavnika v preglednico 1 skicirajte del rastline, ki je pomemben znak za prepoznavo.

**REZULTATI:**

*Preglednica 1: Poimenovanje drevesne vrste in skicirani rastlinski deli.*






Drevesna vrsta	Rastlinski del (list, iglica, vejica)
slovensko ime: _____	
slovensko ime: _____	
slovensko ime: _____	
slovensko ime: _____	
slovensko ime: _____	
slovensko ime: _____	

**NADALJEVANJE:** S pomočjo znanja, pridobljenega na delovni postaji, pridobi namig za naslednjo delovno postajo.

S pomočjo naslednjih namigov ugotovi drevo. Na barvnem kartonu boš našel namig, kje nadaljuješ svojo pot (številko postaje). To drevo **jeseni odvrže vse iglice**. Zanimivo je, da je njegova smola vedno tekoča. Hitro se prepozna glede na barvo cvetov. Moški cvetovi so rumeni, ženska socvetja pa rdeče vijolična. Sicer spada v družino **borovk**, vendar glede na njegovo razporeditev iglic tega ne ugotoviš zlahka. **Iglice se oblikujejo v šopu**.



Priloga A: Slikovni določevalni ključ listavcev.

## LISTAVCI

1. <b>Listi v obliki romba.</b> <b>Pelodne mačice.</b>		<b>BREZA</b>
2. <b>Beli cvetovi.</b> Listi jajčasti. Pri dnu listne ploskve <b>rdeče žleze</b> .		<b>ČEŠNJA</b>
3. <b>Beli cvetovi v dolgih grozdih.</b> Jajčno do ovalni listi.		<b>ČREMSA</b>
4. <b>Lihoperni listi</b> (3, 5 ali 7 lističev). Dva krilata oreška. <b>Zgornja ploskev zelena, spodnja pa svetla.</b>		<b>JAVOR</b>
6. Ovalni do jajčasti listi. Valovit listni rob. Listni rob <b>odlačen</b> .		<b>BUKEV</b>

Priloga B: Slikovni določevalni ključ iglavcev.

## IGLAVCI

1. Iglice v <b>šopu</b> .		<b>MACESEN</b>
2. Po <b>2 iglici skupaj</b> .		<b>BOR</b>
3. Iglice nameščene <b>okrog in okrog vejice</b> . Viseči storži. <b>Koničaste iglice</b> .		<b>SMREKA</b>

4. Igllice nameščene v ravnini.  
Stoječi storži.  
Zaobljene konice iglic.  
Dve svetli progi na spodnji strani iglic.



JELKA

**Priloga 4:** Delovni list za postajo *Kako določimo starost drevesa?*

## Kako določimo starost drevesa?

Ste že kdaj poskusili določiti starost drevesa? Če še niste, se boste tega lotili tokrat.

Ko je na voljo prečen prerez debla, lahko preštejemo **letnice**. To so meje med branikami oz. letnimi prirastki. Iz teh lahko razberemo, v kakšnih pogojih je rastlina rastla, saj so nekateri prirastki širši kot drugi. Letnice pomenijo zimski čas, ko rast dreves ni intenzivna.

Kadar prerez ni na voljo, si lahko pomagamo z **merjenjem obsega** in **izračunom letnega prirastka**. Obseg dreves v gozdu naj bi se v enem letu povprečno povečal za približno 2,5 cm.

**MATERIAL:** Krojaški meter, prečen prerez debla ali štor z vidnimi letnicami.

**NALOGA:** Določite starost drevesa:

1. s pomočjo **štetja letnic** (slika 2),
2. z **merjenjem obsega debla** (slika 1), pri čemer vrednost delite s 2,5.  
Rezultate vpišite v preglednico 1.



Slika 1: Določitev starosti drevesa z merjenjem obsega debla. Slika 2: Določitev starosti drevesa s pomočjo štetja letnic.

### REZULTATI:

Preglednica 1: Meritve starosti drevesa.

Metoda	Število letnic	Merjenje obsega
starost drevesa 1		
starost drevesa 2		

**VPRAŠANJA:**

1. Na skici je prikazan prerez debla. Označite letnico in braniko.



2. V skupini se pogovorite in primerjajte obe metodi določanja starosti dreves. Ob zapisanih trditvah obkrožite metodo, za katero **menite**, da zapisana trditev bolj velja.

Menimo, da:

– je <b>hitrejši</b> način določitve starosti drevesa z metodo:	štetja letnic,	merjenja obsega.
– je <b>natančnejši</b> način določitve starosti drevesa z metodo:	štetja letnic,	merjenja obsega.
– je <b>enostavnejši</b> način določitve starosti drevesa z metodo:	štetja letnic,	merjenja obsega.
– je <b>večja možnost človeške napake</b> pri uporabi metode:	štetja letnic,	merjenja obsega.
– z uporabo te metode <b>manj poškodujemo drevo</b> .	štetje letnic,	merjenje obsega.
– smo z uporabo te metode <b>manj omejeni, kje</b> določujemo starost drevesa (v parku, učilnici idr.).	štetje letnic,	merjenje obsega.

**NADALJEVANJE:** S pomočjo znanja, pridobljenega na delovni postaji, pridobi namig za naslednjo postajo.

Za določevanje starosti drevesa lahko ob prečno prerezanem kosu drevesa uporabimo tudi štor. V parku poišči štor, ki je 41 let star. V njegovi bližini bo tvoja naslednja delovna postaja.

## Priloga 5: Delovni list za postajo *Mozaik listne ploskve*

### Mozaik listne ploskve



Si se kdaj spraševal, kakšen je videz lista v notranjosti? Pogled skozi leče mikroskopa ti prikaže velike skrivnosti o zgradbi rastlin. Kje poteka fotosinteza, s čim rastlina uravnava količino vode in še veliko več. Danes si boš pogledal strukturo lista, ki je zelo pomembna pri teh procesih.

List je rastlinski organ, v katerem poteka proces **fotosinteze** in izmenjave plinov. Za potek fotosinteze morajo biti v celicah prisotni **kloroplasti** z zelenim barvilom klorofil. Za proces fotosinteze so potrebni plini, ki se morajo izmenjevati med rastlino in okoljem. Izmenjava poteka skozi **listnih rež** (slika desno). Te celične strukture se lahko odprejo in tako omogočijo pretok plinov in vode. Kadar pa so zaprte, je pretok onemogočen.

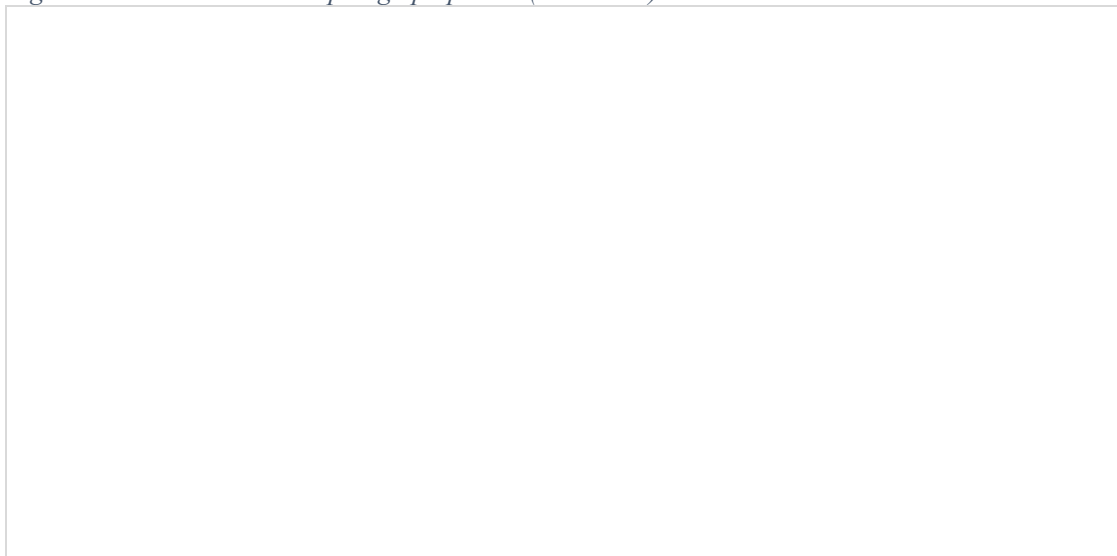


**MATERIAL:** Mikroskop, pribor za mikroskopiranje, rastlina (listi), prozoren lak za nohte.

**NALOGA:** Pripraviti mikroskopski preparat za opazovanje listnih rež. Le-tega pogledati skozi mikroskop in v preglednico 1 skicirati videne celične strukture.

**REZULTATI:**

*Preglednica 1: Skica mikroskopskega preparata (listne reže).*



**NADALJEVANJE:** S pomočjo znanja, pridobljenega na delovni postaji, pridobi namig za naslednjo postajo.

Po uporabi mikroskopa si v poznavanju njegovih delov strokovnjak. Zato poišči **objektiv** in razkrija se ti bo številka naslednje delovne postaje.



**Priloga 6:** Delovni list za postajo *Semena in plodovi*

## Semena in plodovi

V gozdovih, na njihovih obronkih, poljih in travnikih najdemo veliko plodov in semen. Le zakaj jih rastline potrebujejo in kdo se vse z njimi hrani?

Po oploditvi cveta se začne razvoj semena in ploda. **Seme** se razvija v semenski zasnovi, ki je zavarovana s plodnico. Plodnica dozoreva v **plod**. Rastline so različnih barv, oblik in imajo različne velikosti plodov. Dozoreli plodovi poskrbijo za širjenje semen. Številna majhna semena razširjajo **živali, veter, voda, ljudje**, le redke rastline pa semena kar **same izstrelijo**.

**NALOGA 1:** Sestavi sestavljanko, na kateri je zapisano navodilo za nadaljnjo delo (preverjanje prisotnosti olj in škroba v semenih).

### VSEBUJEJO SEMENA OLJA?

**NAVODILO:** V semenih boste preverili prisotnost olj.

**MATERIAL:**

- Različna semena (fižol, sončnica, pšenica itd.),
- barven papir.

**POSTOPEK – POSKUS 1:**

1. Odstranite lupino semena.
2. Seme podrgnite po temnem papirju.
3. Opažanja zapišite v preglednico 1!

### VSEBUJEJO SEMENA ŠKROB?

**NAVODILO:** V semenih boste preverili prisotnost škroba.

**MATERIAL:**

- Različna semena (fižol, sončnica, nektarina idr.),
- jodovica,
- kapalka,
- krožnik ali druga podlaga.

**POSTOPEK – POSKUS 2:**

1. Prepolovite nekaj semen.
2. Semena položite na krožnik ali drugo podlago.
3. Na semena kapnite jodovico.
4. Opažanja zapiši v preglednico 1.

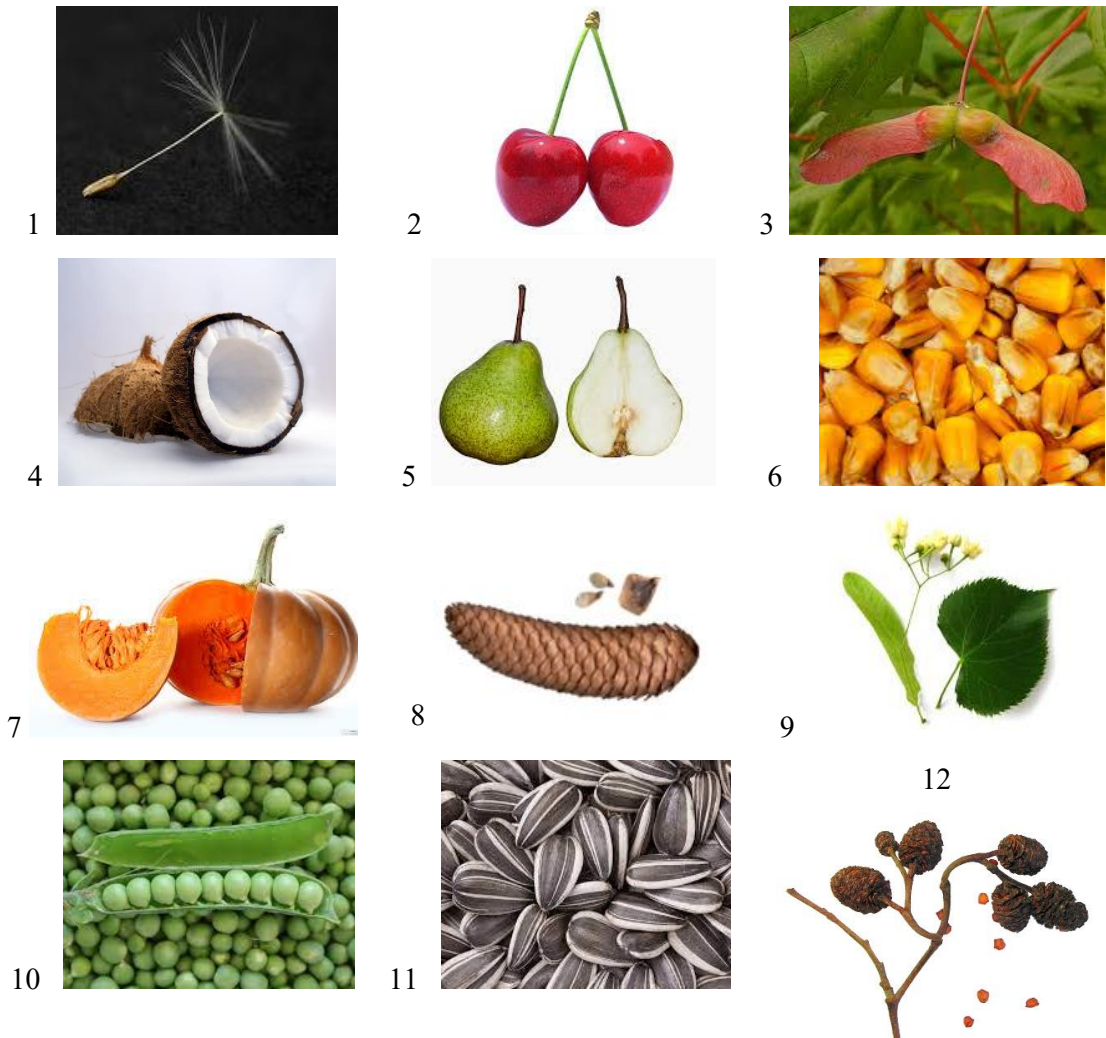


### **REZULTATI:**

*Preglednica 1: Rezultati poskusa 1 in poskusa 2.*

Seme	Prisotnost škroba		Prisotnost olj	
	DA	NE	DA	NE
	DA	NE	DA	NE
	DA	NE	DA	NE
	DA	NE	DA	NE
	DA	NE	DA	NE
	DA	NE	DA	NE

**NALOGA 2:** S številkami 1–12 so označena semena in plodovi. V preglednico 2 jih razporedite glede na to, kdo ali kaj semena oz. plodove razširja.



*Preglednica 2: Semena in plodovi se razširjajo na različne načine.*

S pomočjo <b>vode</b> .	S pomočjo <b>vetra</b> .
S pomočjo <b>živali</b> .	S pomočjo <b>človeka</b> .

**NADALJEVANJE:** S pomočjo znanja, pridobljenega na postaji, pridobi namig za naslednjo postajo. Med pripravami delavnice so se nama zmešala semena. Pomagaj nama sortirati semena v skupine glede na to, kateri rastlini pripadajo. Število bučnih semen ti bo namignilo številko naslednje postaje.

**Priloga 7:** Delovni list za postajo *Ali olje res plava na vodi?***Ali res olje plava na vodi?**

Ste kdaj razmišljali, katere lastnosti vplivajo na uporabnost snovi? Katere lastnosti lahko vse proučujemo? Predvsem zakaj se nekatere kapljevine zadržujejo ena nad drugo? Z znanjem o lastnosti snoveh boste primerjali snovi po določeni lastnosti. Kateri? **Gostoti snovi.**

Gostejše snovi imajo enako maso razporejeno v manjši prostornini. To pomeni, da ima enako število delcev snovi na voljo manjši prostor. Z drugimi besedami povedano nam gostota pove kolikšno maso ima dana prostornina snovi.



**MATERIAL:** Voda, olje, led, kozarci za vlaganje, pomaranča, tehtnica.

**NALOGA:** Narediti poskuse, s katerimi preverimo gostoto različnih snovi.

**POSKUS 1: OLJE in VODA**

1. V manjši kozarec za vlaganje vlijte **do polovice** vodo.
  2. V kozarec dodajte **za debelino palca olje**.
  3. V shemo na desni vrišite plasti olja in vode.
- ❓ Kaj menite, da bi se zgodilo, če bi v kozarec dodali kocko ledu? (*Obkrožite.*)
- a. Led bi plaval na olju.
  - b. Led bi potonil.
  - c. Led bi plaval med vodo in oljem.
4. V kozarec **dodajte kocko ledu**. Preverite vaše ❓ predvidevanje.

**POSKUS 2: IZMERIMO GOSTOTO POMARANČE**

1. S pomočjo tehtnice **stehajte pomarančo** in vrednost zapišite na list.  
m (pomaranče) = \_\_\_\_\_ g
2. V merilno posodo **vlijte vodo** do oznake ½ L.
3. V to posodo **dodajte pomarančo** (Ta mora biti v vodi v celoti potopljena!) in **odčitajte vrednost** na merilni posodi, do kod seže nivo vode.
4. **Izračunajte razliko**, za koliko se je povišal nivo gladine vode. Ta številka predstavlja prostornino pomaranče.



$$V (\text{pomaranče}) = \text{_____ mL} - 500 \text{ mL} = \text{_____ mL}$$

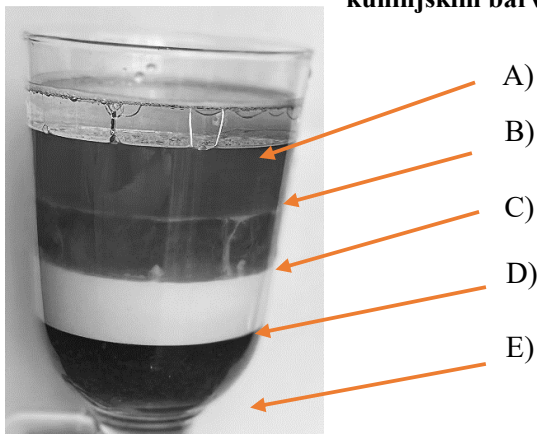
5. Iz mase in prostornine izračunajte gostoto pomaranče. To storite tako, da v zapisano enačbo vstavite vrednosti.

$$\rho (\text{pomaranče}) = \frac{m (\text{pomaranče})}{V (\text{pomaranče})} = \frac{\text{_____ g}}{\text{_____ mL}} = \frac{\text{_____ g}}{\text{_____ mL}}$$

### POSKUS 3: TEKOČINSKI STOLPEC

V kozarec sva predhodno nalili v **pravilnem zaporedju** kapljevine, tako kot prikazuje fotografija na desni. Tekočinski stolpec sva fotografirali, vendar sva prepozno ugotovili, da imava zgolj fotografijo v različnih odtenkih sive.

Poskusite ugotoviti in zapišite, v kakšnem vrstnem redu si sledijo kapljevine, če veste, da sva uporabili **mleko, med, detergent za pomivanje posode, olje in vodo**, ki sva jo rahlo obarvali z rdečim kuhinjskim barvilom.



Ko ob zgornje črke zapišete snovi, **poskenirajte QR kodo** in **preverite** vaše predhodno sklepanje.



NADALJEVANJE: S pomočjo znanja, pridobljenega na postaji, pridobi namig za naslednjo postajo. Številka naslednje postaje se skriva v tekočinskem stolpcu. Poišči rdečo obarvano tekočino in ugotovi, **katera zapovrstjo** je bila dodana v kozarec.

# Akademija paralelnega algoritma razveji in omeji za 0/1 nahrbtnik in delo s sistemom Singularity

## Guide of parallel branch and bound algorithm for the 0/1 knapsack problem and work with the Singularity system

Nik Hrastnik, Aleksander Bajc

*Univerza v Mariboru, Fakulteta za naravoslovje in matematiko, Koroška cesta 160, 2000 Maribor, Slovenija*

---

### Povzetek

V tej akademiji s postopnim reševanjem nalog naučimo osnov paralelnega reševanja optimizacijskih problemov na primeru 0/1 nahrbtnika. Najprej spoznamo matematično formulacijo problema 0/1 nahrbtnika, katerega cilj je izbrati optimalno kombinacijo predmetov glede na njihovo težo in vrednost ob omejeni kapaciteti nahrbtnika. Nato uvedemo standard MPI za paralelno programiranje, ki omogoča sodelovanje več procesov pri reševanju istega problema z izmenjavo sporočil. Pri iskanju optimalne rešitve uporabimo metodo razveji in omeji, ki zaradi svoje strukture omogoča učinkovito paralelizacijo. Na koncu se naučimo program zagnati tudi na superračunalniku ter spoznamo uporabo platforme za vsebnike Singularity, ki omogoča prenosljivo in ponovljivo izvajanje programov.

*Ključne besede: Nahrbtnik 0/1, MPI, HPC, Python, Singularity.*

### Abstract

In this guide, we gradually learn the fundamentals of parallel solving of optimisation problems using the 0/1 knapsack problem as case study. We begin by introducing the mathematical formulation of the 0/1 knapsack problem, whose goal is to select an optimal combination of items based on their weights and values under a limited knapsack capacity. We then introduce the MPI standard for parallel programming, which enables multiple processes to cooperate in solving the same problem through message passing. To find the optimal solution, we employ the branch and bound method, which, due to its structure, allows for efficient parallelization. Finally, we learn how to run the program on a supercomputer and explore the use of the Singularity container system for portable and reproducible execution of programs.

*Key words: Knapsack 0/1, MPI, HPC, Python, Singularity.*

---

## 1 Uvod

Problem 0/1 nahrbtnika sodi med klasične optimizacijske probleme, pri katerih želimo iz množice predmetov izbrati takšne, ki jih lahko spravimo v nahrbtnik omejene kapacitete, obenem pa maksimiziramo skupno vrednost izbranih predmetov. Vsak predmet ima podano težo in vrednost. Za vsakega se odločimo, da ga damo oz. da ga ne damo v nahrbtnik.

Podanih imamo  $n$  predmetov, ki so oštevilčeni od 1 do  $n$ . Za vsak  $i \in \{1, \dots, n\}$  imamo podano vrednost  $c_i$ , ki predstavlja ceno oz. vrednost predmeta, in  $w_i$ , ki predstavlja težo

predmeta. Podana je tudi maksimalna teža nahrbtnika  $W$ . Naša naloga je poiskati

$$\max \left\{ \sum_{i=1}^n c_i x_i \mid \sum_{i=1}^n w_i x_i \leq W, x_i \in \{0, 1\} \right\}$$

[3]. Za reševanje obravnavanega problema bomo uporabili MPI (angl. *Message Passing Interface*), uveljavljeni standard za paralelno programiranje, ki omogoča sodelovanje več procesov z izmenjavo sporočil. Takšen pristop omogoča učinkovito porazdelitev računskih nalog ter koordinacijo procesov pri iskanju optimalne rešitve [5].

Problema 0/1 nahrbtnika bo reševan z metodo razveji in omeji (angl. *branch and bound*), ki sistematično pregleduje prostor možnih rešitev tako, da problem razdeli na manjše podprobleme. Pri tem se uporabljajo zgornje in spodnje meje za izločanje tistih vej iskalnega drevesa, za katere je mogoče dokazati, da ne morejo voditi do optimalne rešitve. Zaradi neodvisnosti posameznih vej iskalnega drevesa je metoda posebej primerna za paralelno izvajanje, saj lahko ločeni procesi hkrati obravnavajo različne dele iskalnega prostora ter si izmenjujejo informacije o trenutno najboljši znani rešitvi.

Algoritem razveji in omeji deluje na naslednji način. Problem rešujemo s pomočjo iskalnega drevesa. Ko smo v drevesu na nivoju  $i$ , se odločamo, ali bomo predmet  $i$  vzeli v nahrbtnik ali ne. Tako ima vsako vozlišče dva otroka. Pri enem smo predmet  $i$  dodali v nahrbtnik, pri drugem pa ne. V vsakem vozlišču sproti tudi shranjujemo seštevek tež in cen do takrat izbranih predmetov. Če teža na neki točki preseže maksimalno težo, te veje več ne raziskujemo naprej.

V vsakem vozlišču izračunamo tudi zgornjo mejo mogoče vrednosti, ki jo še lahko dosežemo s trenutnim izborom predmetov. To naredimo tako, da vzamemo vse nadaljnje predmete, kolikor jih še lahko, ter del naslednjega predmeta, če še kateri ostane. Če je ta zgornja meja manjša ali enaka najboljši do sedaj znani rešitvi, te veje več ne raziskujemo naprej, saj z njo ne moremo priti do optimalne rešitve. V nasprotnem primeru drevo iz tega vozlišča razvijemo naprej z dvema otrokoma, kot je že bilo opisano. Po končani obdelavi drevesa optimalno rešitev najdemo na zadnjem nivoju. To bo vozlišče z najvišjo ceno in ustrezno težo [2].

Ko bo program za nahrbtnik deloval na naših računalnikih, se ga bomo naučili pognati še na superračunalniku (HPC - high performance computing), potem pa bomo zanj zgradili še vsebnik z uporabo platforme Singularity.

Singularity (danes znan tudi kot Apptainer) je odprtokodni sistem za izvajanje vsebnikov, zasnovan posebej za potrebe znanstvenega računalništva in visokozmogljivega računanja (HPC). V nasprotju z drugimi vsebniki tehnologijami, kot je Docker, je Singularity prilagojen okoljem, kjer uporabniki nimajo administratorskih privilegijev. Vsebniki v Singularityju omogočajo, da raziskovalci zapakirajo celotno programsko okolje – knjižnice, pakete in odvisnosti – v prenosljivo enoto, ki se potem izvede na HPC sistemih [6].

## 2 Naloge

### 2.1 Osnovno o MPI

Da bomo lahko implementirali 0/1 nahrbtnik s pomočjo MPI, se moramo najprej poučiti o osnovnih MPI funkcijah.

- Pozanimaj se o pomenu MPI spremenljivke `MPI.COMM_WORLD` in metod `comm.Get_rank()` in `comm.Get_size()`.
- Pozanimaj se še o pomenu preostalih metod, ki jih bomo potrebovali v programu, kot so `comm.send()`, `comm.recv` in `comm.IProbe()`.

## 2.2 Implementacija 0/1 nahrbtnika v Pythonu z MPI

### 2.2.1 Ustvarjanje osnovnih razredov

Kot je že zapisano, bomo za našo rešitev uporabili metodo razveji in omeji, ki za iskanje rešitev uporablja drevesa. Zato bomo v Pythonu potrebovali razred `Vozlišče`. Pri vsakem vozlišču bomo imeli dva otroka, enega za primer, ko dani predmet vzamemo v nahrbtnik, drugega pa za primer, ko predmeta ne vzamemo. Zato bomo v vsakem vozlišču tudi hranili trenutno stanje nahrbtnika v obliki polja. Prav tako pa potrebujemo razred `Element` za predstavitev vseh predmetov, ki jih imamo na voljo za nahrbtnik.

- Definirajte razred `Element`, ki predstavlja predmet z atributi `cena`, `teza` in `index`. Dodajte tudi izračun razmerja `cena/teza`.
- Definirajte razred `Vozlisce`, ki predstavlja vozlišče v iskalnem drevesu. Naj vsebuje attribute `nivo`, `cena`, `teza`, `nahrbtnik` in `ocena`.

### 2.2.2 Urejanje predmetov glede na razmerje cena/teža

Za učinkovito metodo razveji in omeji bomo predmete najprej uredili glede na razmerje `cena/teža`.

- Implementirajte funkcijo `uredi_elemente(predmeti)`, ki vrne seznam predmetov, urejen po padajočem razmerju `cena/teza`.

### 2.2.3 Zapis funkcije za izračun zgornje meje cene

Metoda razveji in omeji uporablja funkcijo zgornje meje, ki oceni, kakšna je lahko maksimalna teoretična vrednost, če bi od trenutnega vozlišča naprej najprej dodali vse predmete, ki celi še gredo v nahrbtnik, nato pa še del naslednjega predmeta, kolikor ga gre v nahrbtnik. To je tudi razlog, zakaj smo pri prejšnji nalogi uredili predmete glede na razmerje `cena/teza`.

- Implementirajte funkcijo `zgornja_meja_cene(v, predmeti, max_teza)`.
- Če vozlišče že presega največjo dovoljeno težo, naj funkcija vrne 0.
- Funkcija naj vrne maksimalno možno ceno, ki jo lahko teoretično dosežemo iz danega vozlišča, če dovolimo dodajanje delnega zadnjega predmeta. Dokler ne presežemo maksimalne teže, dodajamo predmete, na koncu pa dodamo še del naslednjega predmeta, kolikor gre.

### 2.2.4 Zapis algoritma razveji in omeji (mojster proces)

Sedaj bomo zapisali algoritem razveji in omeji, ki je glavni algoritem za rešitev našega problema. Zato definirajte funkcijo, ki za parametra dobi predmete in maksimalno težo. Vso kodo v tej in naslednji nalogi pišite znotraj te funkcije.

V tej nalogi boste razvili del kode, ki teče na mojster procesu. Mojster upravlja z delovnimi procesi (delavci), deli delo, zbira rezultate in skrbi za globalno najboljšo rešitev. Preden začnemo s pisanjem kode, razložimo še nekaj osnovnih pojmov, ki jih boste potrebovali za razumevanje.

Pri paralelnem programiranju z uporabo knjižnice MPI uporabljamo komunikator, ki predstavlja skupino procesov, med katerimi poteka komunikacija. Najpogosteje se uporablja komunikator `COMM_WORLD`, ki vključuje vse procese, sodelujoče pri izvajanju programa.

Spremenljivka `rank` označuje enolični identifikator posameznega procesa znotraj komunikatorja. Na podlagi vrednosti `rank` ločimo mojster proces (`rank = 0`) od delavskih procesov (`rank > 0`), ki izvajajo dodeljene naloge.

Spremenljivka `size` predstavlja skupno število procesov, ki sodelujejo pri izvajanju programa. Ta informacija je pomembna pri razdeljevanju dela med procese.

1. Uvozite knjižnico MPI iz `mpi4py` in inicializirajte MPI okolje:
  - pridobite komunikator `COMM_WORLD`,
  - ugotovite `rank` in `size`.
2. Uredite seznam predmetov po koristnosti (cena/teža) in shranite število predmetov v spremenljivko.
3. Določite ključne spremenljivke:
  - `STOP_SIGNAL` za končanje delavcev na `None`,
  - `max_cena_global` za globalno najboljšo rešitev na `0`,
  - `max_nahrbtnik_global` za globalno najboljšo kombinacijo predmetov na prazno polje.
4. Zapišite `if` stavek, ki bo preveril, da smo v procesu z rangom `0`.
5. Ustvarite podatkovno strukturo vrsta `deque` za obdelavo vozlišč. Gre za dvostransko vrsto, ki omogoča učinkovito dodajanje in odstranjevanje elementov tako na začetku kot na koncu. Za to boste potrebovali knjižnico `deque` iz `collections`.
6. Inicializirajte začetno vozlišče:
  - `koren = Vozlisce()`,
  - uporabite prej zapisano funkcijo za izračun zgornje meje in jo shranite v atribut `ocena` tega vozlišča,
  - dodajte ga v vrsto.
7. Uvedite števec aktivnih delavcev in ga nastavite na `0`.
8. Začnite glavno zanko, ki se izvaja, dokler:
  - obstajajo neobdelana vozlišča v vrsti in
  - še obstaja kak prosti delavec.
9. Če je v vrsti vozlišče in imamo prostega delavca:
  - iz vrste odstranite prvo vozlišče,
  - določite ciljnega delavca,

- pošljite vozlišče za obdelavo temu delavcu s `comm.send()`. Uporabite parameter `tag` in mu določite vrednost 1. Tako oznako bodo imele naloge, ki jih mojster pošilja delavcem. Oznako 2 bodo imeli rezultati, ki jih bo mojster prejemal od delavcev.
  - povečajte število aktivnih delavcev.
10. Z uporabo `comm.Iprobe` preverite, ali je prispelo sporočilo z oznako `tag = 2`.
11. Če je sporočilo prispelo:
- pridobite `worker_id` iz `status` objekta,
  - prejmite rezultat s pomočjo `comm.recv`,
  - zmanjšajte število aktivnih delavcev,
  - pridobite lokalno najboljšo rešitev in nova vozlišča iz rezultata,
  - po potrebi posodobite globalno najboljšo rešitev.
12. Vsako vrnjeno vozlišče dodajte nazaj v čakalno vrsto, če njegova ocena presega trenutno globalno najboljšo ceno. Ta vozlišča se še splača pregledati naprej.
13. Ko zmanjka dela in so vsi delavci prosti:
- pošljite vsem delavcem `STOP_SIGNAL` s `tag = 0`,
  - prekinite glavno zanko.
14. Izpišite končno najboljšo globalno rešitev:
- maksimalno ceno,
  - nahrbtnik v originalnem vrstnem redu predmetov.

### 2.2.5 Zapis algoritma razveji in omeji (delovni proces)

Še vedno ostajamo znotraj funkcije za algoritem razveji in omeji, definirane v prejšnji nalogi. V tej nalogi boste implementirali del kode, ki ga izvajajo delovni procesi (tisti z `rank > 0`). Delavec prejme posamezno vozlišče, ga obdelava in rezultate pošlje nazaj mojstru.

1. Začnite z `else` stavkom, ki se navezuje na `if rank == 0` iz prejšnje naloge. Pridobite `rank` trenutnega delavca in začnite neskončno zanko, ki se konča šele, ko prejmete `STOP_SIGNAL`.
2. Prejmite vozlišče od mojstra s funkcijo `comm.recv()`. Če je prejeto vozlišče `None`, prekinite zanko (delavec se ustavi).
3. Inicializirajte lokalne spremenljivke:
  - `lokalna_cena = 0`,
  - `lokalni_nahrbtnik = []`,
  - `nova_vozlisca = []`.
4. Če vozlišče še ni na zadnjem nivoju drevesa, mu bomo ustvarili dva otroka. Pri enem otroku bomo trenutni predmet vzeli v nahrbtnik, pri drugem pa ne:

- Ustvarite otroka **v1**, ki predmet vzame:
  - povečajte nivo za 1,
  - posodobite težo in ceno s težo in ceno trenutnega predmeta,
  - v seznam nahrbtnika dodajte 1.
- Če teža ne presega maksimalne in je cena boljša od trenutne lokalne cene, posodobite najboljšo lokalno rešitev.
- Izračunajte zgornjo mejo vrednosti za **v1** in jo shranite v atribut `ocena` tega vozlišča.
- Dodajte **v1** v seznam novih vozlišč.
- Ustvarite otroka **v2**, ki predmeta ne vzame:
  - povečajte nivo za 1,
  - ohranite težo in ceno,
  - dodajte 0 v nahrbtnik.
- Izračunajte zgornjo mejo vrednosti za **v2**.
- Dodajte **v2** v seznam novih vozlišč.

5. Po koncu tega razvejanja pošljite mojstru listo podatkov:

```
(lokalna_cena, lokalni_nahrbtnik, nova_vozlisca)
```

z oznako `tag = 2`.

### 2.2.6 Zagon programa na preprostem primeru

Zapisan program bomo preizkusili na preprostem primeru.

- V glavni funkciji definirajte polje `predmeti`. Prvi predmet naj ima ceno 10 in težo 4, drugi predmet ceno 6 in težo 6 ter tretji predmet ceno 12 in težo 9. Definirajte spremenljivko za maksimalno težo in jo določite na 15. Kličite funkcijo za algoritem razveji in omeji za definirane predmete in maksimalno težo.
- Kodo boste zagnali znotraj `Command prompt` programa, ki ga odprete na vašem računalniku.
- Če še v vašem Python okolju nimate nameščene knjižnice `mpi4py`, jo namestite iz spletne strani Microsofta, potem pa še znotraj konzole zaženete ukaz `pip install mpi4py`.
- Z ukazom `cd` in zapisom celotne poti se premaknete v mapo, kjer se nahaja Python skripta.
- Program zaženite z uporabo `mpirexec` in uporabite 4 različne procese.

### 2.2.7 Zagon programa na HPC

Sedaj bomo zapisan program zagnali na superračunalniku.

- Pridobite si dostop in se povežite na superračunalnik. Sledite navodilom v članku *Navigacija po svetu superračunalništva: Praktični vodnik za prijavo projekta, dostop in uporabo gručnega okolja* na povezavi [https://www.fnm.um.si/wp-content/uploads/2025/09/Dianoia\\_2025\\_2.pdf](https://www.fnm.um.si/wp-content/uploads/2025/09/Dianoia_2025_2.pdf).

- Na HPC se uporablja Slurm, ki je sistemski upravljalnik opravil. Zadolžen je za razporejanje in nadzor izvajanja programov. Da lahko na HPC zaženemo naš program, moramo računalniku najprej dati določena navodila v zvezi z izvedbo, recimo koliko vozlišč naj se uporabi za naš program. Ta navodila zapišemo v datoteko s končnico `slurm`. Ustvarite `slurm` datoteko in vanjo prilepite spodnjo kodo:

```
#!/bin/bash
#SBATCH --job-name=nahrbtnik
#SBATCH --output=nahrbtnik_%j.out
#SBATCH --error=nahrbtnik_%j.err
#SBATCH --time=00:05:00
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=2
#SBATCH --partition=cpu
```

V njej so podatki o imenu naloge, o imenu out datoteke, kjer bodo izpisani rezultati programa, o imenu error datoteke, kjer bodo izpisane napake, do katerih je prišlo med izvedbo programa, in podatki v zvezi z vozlišči, ki jih potrebujemo za izvedbo programa.

- Ker bomo potrebovali knjižnico za MPI, v našo skripto dodamo Anaconda, ki je na HPC okolje za upravljanje Python paketov. V skripto dodajte naslednje:

```
module load Anaconda3
module load OpenMPI/5.0.3-GCC-13.3.0

# Aktiviraj conda okolje, kjer bomo imeli mpi4py
source $(conda info --base)/etc/profile.d/conda.sh$
conda activate mpi_env

echo "Job started"

srun --mpi=pmi2 python -m mpi4py nahrbtnik.py
```

Pri tem pazite, da v zadnji vrstici uporabite ime svoje datoteke s končnico `py`.

- Naložite ustrezni datoteki s končnicama `py` in `slurm` na WinSCP in se prijavite v HPC. Najprej se z ukazom `cd` prestavite v svojo mapo na WinSCP, v kateri imate datoteko s kodo.
- V HPC zapišite enake ukaze, kot ste jih pred tem v skripto:

```
module load Anaconda3
module load OpenMPI/5.0.3-GCC-13.3.0
source $(conda info --base)/etc/profile.d/conda.sh$
conda activate mpi_env
```

- Zaženite `slurm` skripto z ukazom `sbatch nahrbtnik.slurm`.
- Počakajte, dokler se v vaši mapi ne pojavita datoteki `nahrbtnik_.out` in `nahrbtnik_.err`. Kaj je v vaši mapi, lahko preverite z ukazom `ls`.

- Zaženite vaš program z ukazom `cat nahrbtnik_.out`. Če rezultati niso ustrezni ali pa se program ne izvede, lahko preverite napake v vaši kodi z ukazom `cat nahrbtnik_.err`.

## 2.3 Delo s Singularity

### 2.3.1 Osnovno o Singularity

- Pozanimaj se o pomenu vsebnikov Singularity in kaj so prednosti uporabe le-teh v HPC okolju.
- Pozanimaj se, katere komponente mora vsebovati vsebnik Singularity, da lahko zažene MPI Python program.

### 2.3.2 Priprava vsebnika

- Kaj je SIF datoteka in kakšna je njena vloga pri uporabi vsebnikov Singularity?
- Opiši postopek priprave vsebnika Singularity.

### 2.3.3 Implementacija vsebnika Singularity za problem 0/1 nahrbtnika

Algoritma za reševanje problema 0/1 nahrbtnika ter njegove MPI implementacije ni treba spreminjati. Vsebnik Singularity uporabimo izključno kot tehnično rešitev za zagotavljanje enotnega in zanesljivega programskega okolja na HPC sistemu.

- Za obstoječo MPI Python implementacijo problema 0/1 nahrbtnika pripravi vsebnik Singularity.
  - Preveri, da MPI Python program (`nahrbtnik.py`) deluje lokalno.
  - Ustvari definicijsko datoteko (`nahrbtnik.def`) in v njej določi:
    - \* osnovno Linux sliko,
    - \* Python okolje in potrebne Python pakete,
    - \* MPI knjižnice,
    - \* Python skripto za problem 0/1 nahrbtnika.
  - Zgradi SIF datoteko (`nahrbtnik.sif`).
  - Preizkusi vsebnik lokalno.
  - Prenesi SIF datoteko na HPC sistem.
  - Zaženi MPI program znotraj vsebnika.

## 3 Rešitve

### 3.1 Rešitev za nalogo 2.1

Za pomoč si lahko preberete članek *Navigacija po svetu superračunalništva: Praktični vodnik za prijavo projekta, dostop in uporabo gručnega okolja* na povezavi [https://www.fnm.um.si/wp-content/uploads/2025/09/Dianoia\\_2025\\_2.pdf](https://www.fnm.um.si/wp-content/uploads/2025/09/Dianoia_2025_2.pdf).

S pomočjo `MPI.COMM_WORLD` pridobimo komunikacijski kanal med vsemi procesi, z `Get_rank()` določimo identifikacijsko številko (rang) posameznega procesa, medtem ko `Get_size()` vrne skupno število procesov, ki sodelujejo pri izvajanju programa.

Metoda `comm.send(data, dest)` pošlje izbrane podatke procesu z rangom `dest`, medtem ko `comm.recv(source)` omogoča prejem podatkov od določenega procesa. Metoda `comm.IProbe(source)` omogoča neblokirano preverjanje, ali je na voljo dohodno sporočilo. To pomeni, da se proces ne ustavi, če sporočilo še ni prispelo, temveč lahko nadaljuje z drugimi opravili [1].

### 3.2 Rešitev za nalogo 2.2.1

```
class Element:
    def __init__(self, cena, teza, index):
        self.cena = cena
        self.teza = teza
        self.index = index
        self.razmerje = cena / teza

class Vozlisce:
    def __init__(self, nivo=0, cena=0, teza=0,
nahrbtnik=None, ocena=0):
        self.nivo = nivo
        self.cena = cena
        self.teza = teza
        self.nahrbtnik = nahrbtnik if nahrbtnik
is not None else []
        self.ocena = ocena
```

### 3.3 Rešitev za nalogo 2.2.2

```
def uredi_elemente(predmeti):
    return sorted(predmeti, key=lambda x: x.razmerje,
reverse=True)
```

### 3.4 Rešitev za nalogo 2.2.3

```
def zgornja_meja_cene(v, predmeti, max_teza):
    if v.teza >= max_teza: # če je vozlišče prekoračilo
# max težo, te veje nima smisla nadaljevati
        return 0
    ocena = v.cena
    j = v.nivo
    skupna_teza = v.teza

    # dokler lahko dodajamo cele predmete, jih dodajamo
    while j < len(predmeti) and skupna_teza +
+ predmeti[j].teza <= max_teza:
        skupna_teza += predmeti[j].teza
        ocena += predmeti[j].cena
        j += 1

    # dodamo še del naslednjega predmeta, kolikor gre
```

```

if j < len(predmeti):
    preostalo = max_teza - skupna_teza
    ocena += predmeti[j].cena *
        * (preostalo / predmeti[j].teza)
return ocena

```

### 3.5 Rešitev za nalogo 2.2.4

```

from mpi4py import MPI
from collections import deque

# razveji in omeji za 0/1 nahrbtnik
def branch_and_bound_nahrbtnik(
    predmeti, max_teza):

    comm = MPI.COMM_WORLD
    rank = comm.Get_rank()
    size = comm.Get_size()

    predmeti = uredi_elemente(predmeti)
    n = len(predmeti)

    STOP_SIGNAL = None
    max_cena_global = 0
    max_nahrbtnik_global = []

    # mojster proces
    if rank == 0:
        vrsta = deque()
        koren = Vozlisce()
        koren.ocena = zgornja_meja_cene(
            koren, predmeti, max_teza)
        vrsta.append(koren)

        aktivni_workerji = 0

        # mojster deli delo, dokler
        # imamo kaj narediti
        while True:

            # če imamo delo, ga pošljemo
            # prostemu delavcu
            if vrsta and \
                aktivni_workerji < size - 1:

                vozlisce = vrsta.popleft()

                dest = (aktivni_workerji %
                    (size - 1)) + 1

```

```
comm.send(
    vozlisce,
    dest=dest,
    tag=1
)

aktivni_workerji += 1

# sprejmemo rezultate
status = MPI.Status()
if comm.Iprobe(
    source=MPI.ANY_SOURCE,
    tag=2,
    status=status):

    worker_id = \
        status.Get_source()

    rezultat = comm.recv(
        source=worker_id,
        tag=2
    )

    aktivni_workerji -= 1

    # posodobimo najboljšo rešitev
    lokalna_cena, \
    lokalni_nahrbtnik, \
    nova_vozlisca = rezultat

    if lokalna_cena > \
        max_cena_global:

        max_cena_global = \
            lokalna_cena

        max_nahrbtnik_global = \
            lokalni_nahrbtnik + \
            [0] * (n -
                len(lokalni_nahrbtnik))

    # dodamo nova vozlišča v vrsto
    for nv in nova_vozlisca:
        if nv.ocena > \
            max_cena_global:

            vrsta.append(nv)
```

```

# končamo, če ni dela
if not vrsta and \
    aktivni_workerji == 0:

    for i in range(1, size):
        comm.send(
            STOP_SIGNAL,
            dest=i,
            tag=0
        )
    break

print("Najvecja cena =",
      max_cena_global)

nahrbtnik_original = [0] * n
for pos, value in \
    enumerate(
        max_nahrbtnik_global):

    orig_index = \
        predmeti[pos].index

    nahrbtnik_original[
        orig_index] = value

print("Nahrbtnik "
      "(originalni "
      "vrstni red):",
      nahrbtnik_original)

```

### 3.6 Rešitev za nalogo 2.2.5

```

# procesi delavcev (rank > 0)
else:
    rank = comm.Get_rank()

    while True:

        vozlisce = comm.recv(
            source=0,
            tag=MPI.ANY_TAG,
            status=MPI.Status()
        )

        if vozlisce is None:
            break

```

```
# inicializiramo lokalne rezultate
lokalna_cena = 0
lokalni_nahrbtnik = []
nova_vozlisca = []

# razvejanje, če nismo v listu
if vozlisce.nivo < len(predmeti):

    # otrok v1: dodamo predmet
    v1 = Vozlisce(
        nivo=vozlisce.nivo + 1,
        teza=vozlisce.teza +
            predmeti[
                vozlisce.nivo].teza,
        cena=vozlisce.cena +
            predmeti[
                vozlisce.nivo].cena,
        nahrbtnik=
            vozlisce.nahrbtnik +
            [1]
    )

    # preverimo težo
    if v1.teza <= max_teza:

        if v1.cena > lokalna_cena:
            lokalna_cena = v1.cena
            lokalni_nahrbtnik = \
                v1.nahrbtnik

        v1.ocena = zgornja_meja_cene(
            v1, predmeti, max_teza
        )
        nova_vozlisca.append(v1)

    # otrok v2: ne dodamo predmeta
    v2 = Vozlisce(
        nivo=vozlisce.nivo + 1,
        teza=vozlisce.teza,
        cena=vozlisce.cena,
        nahrbtnik=
            vozlisce.nahrbtnik +
            [0]
    )
    v2.ocena = zgornja_meja_cene(
        v2, predmeti, max_teza
    )
    nova_vozlisca.append(v2)
```

```
# pošlji rezultat mojstru
comm.send(
    (lokalna_cena,
     lokalni_nahrbtnik,
     nova_vozlisca),
    dest=0,
    tag=2
)
```

### 3.7 Rešitev za nalogo 2.2.6

```
if __name__ == "__main__":
    predmeti = [
        Element(10, 4, 0),
        Element(6, 6, 1),
        Element(12, 9, 2)
    ]
    max_teza = 15
    branch_and_bound_nahrbtnik(predmeti, max_teza)
```

Povezava za nalaganje Microsoft MPI: <https://www.microsoft.com/en-us/download/details.aspx?id=105289>

Prestavimo se v ustrezno mapo:

```
cd "C:\..."
```

Zaženemo program s štirimi procesi:

```
mpiexec -n 4 python mpi_nahrbtnik.py
```

Pričakovani izpis:

```
Najvecja cena = 22
```

```
Nahrbtnik (originalni vrstni red): [1, 0, 1]
```

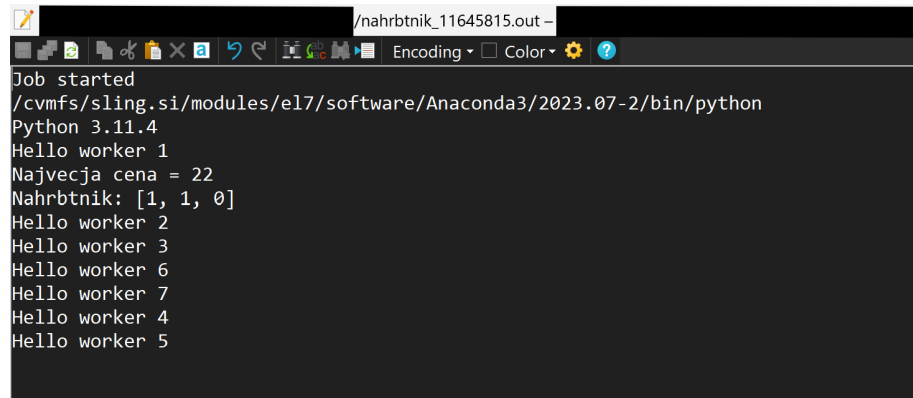
### 3.8 Rešitev za nalogo 2.2.7

Primer izpisa `out` in `err` datoteke vidimo na slikah 1 in 2. V `out` datoteki so poleg rezultatov še izpisi `Hello worker`, ki so bili dodani samo kot dodatno preverjanje, kdaj kateri delavec izvede svoj del naloge. `err` datoteka je prazna, kar pomeni, da pri izvedbi programa ni bilo napak.

### 3.9 Rešitev za nalogo 2.3.1

Osnovno o programski opremi Singularity si lahko preberete v priročniku na povezavi <https://docs.sylabs.io/guides/3.5/user-guide/index.html>.

- Singularity je platforma za ustvarjanje in izvajanje vsebnikov. To so izolirana okolja, ki vsebujejo aplikacijo in vse njene odvisnosti (OS, knjižnice, programske pakete). Razlogi za uporabo Singularity pri delu na HPC:

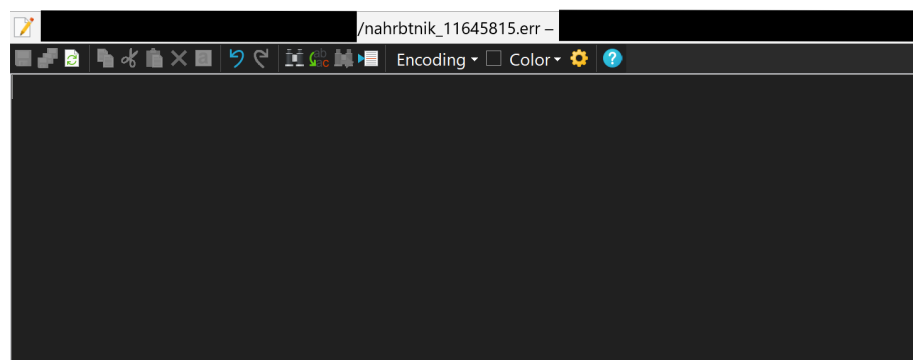


```

/nahrbtnik_11645815.out -
Job started
/cvmfs/sling.si/modules/el7/software/Anaconda3/2023.07-2/bin/python
Python 3.11.4
Hello worker 1
Najvecja cena = 22
Nahrbtnik: [1, 1, 0]
Hello worker 2
Hello worker 3
Hello worker 6
Hello worker 7
Hello worker 4
Hello worker 5

```

Slika 1: Izpis .out datoteke.



```

/nahrbtnik_11645815.err -

```

Slika 2: Izpis .err datoteke.

- **prenosljivost:** program v vsebniku deluje enako na različnih HPC sistemih,
  - **zanesljivost:** rezultati programov so vedno enaki, ne glede na različico OS ali knjižnice HPC sistema,
  - **izolacija:** preprečuje konflikte med različicami Python paketov ali MPI knjižnicami,
  - **MPI podpora:** omogoča zagon MPI programov na več vozliščih brez težav z odvisnostmi,
  - **varno izvajanje:** ne zahteva root dostopa, kar je za nas pomembno, saj na HPC sistemu nimamo root dostopa.
- Komponente, ki jih mora vsebovati vsebnik za MPI Python program:
    - Linux OS in osnovne knjižnice (zagotavlja, da se program izvaja neodvisno od OS HPC-ja),
    - Python in uporabljeni Python paketi,
    - MPI knjižnice (npr. OpenMPI),
    - program, ki ga želimo pognati.

V praksi to pomeni, da vsebnik shranimo v eno datoteko, imenovano SIF (*angl.* Singularity Image File). Ta datoteka vsebuje vse zgoraj navedene komponente. Prenešemo in zaženemo jo lahko na različnih HPC sistemih z minimalno dodatno konfiguracijo [4].

### 3.10 Rešitev za nalogo 2.3.2

- Vsebnik Singularity je shranjen v eni sami datoteki, imenovani **SIF**. SIF datoteka predstavlja celoten vsebnik in vsebuje osnovni operacijski sistem, vse potrebne knjižnice in program, ki ga želimo izvajati.
- Postopek priprave vsebnika Singularity:
  1. Ustvarimo definicijsko datoteko. To je datoteka, običajno s končnico `.def`, ki opisuje, kako je vsebnik zgrajen. Definicijska datoteka vsebuje:
    - osnovno Linux sliko, iz katere izhajamo (npr. Ubuntu),
    - seznam paketov in knjižnic, ki jih bomo uporabljali,
    - Python okolje in potrebne Python pakete,
    - datoteke oziroma programe, ki jih želimo vključiti v vsebnik,
    - ukaz, ki se izvede ob zagonu vsebnika.
  2. Iz definicijske datoteke zgradimo vsebnik z ukazom: `singularity build test_container.sif test_container.def`  
Rezultat tega koraka je SIF datoteka, ki vsebuje celoten vsebnik in vse njegove odvisnosti.
  3. Zgrajeni vsebnik lahko zaženemo z ukazom: `singularity exec test_container.sif python3 nahrbtnik.py`  
Pri MPI programih bomo vsebnik zaganjali v kombinaciji z `mpirun`. Torej: `mpirun -np 4 singularity exec test_container.sif python3 nahrbtnik.py`  
V tem primeru se koda izvaja znotraj vsebnika.

S tem postopkom zagotovimo, da je programsko okolje popolnoma enako na vseh vozliščih HPC sistema, kar poenostavi izvajanje MPI programov in poveča zanesljivost rezultatov [4].

### 3.11 Rešitev za nalogo 2.3.3

- Ustvari definicijsko datoteko `.def`, v kateri določi:
  - osnovno Linux sliko,
  - uporabljeno Python okolje in potrebne Python pakete,
  - MPI knjižnice,
  - Python skripto za reševanje problema 0/1 nahrbtnika.

Primer definicijske datoteke:

```
Bootstrap: docker
From: ubuntu:22.04

%post
    set -eux

    apt-get update && apt-get install -y \
        python3 \
```

```

python3-pip \
python3-dev \
build-essential \
openmpi-bin \
libopenmpi-dev \
&& apt-get clean && rm -rf
    /var/lib/apt/lists/*

pip3 install --upgrade pip
pip3 install mpi4py

rm -rf /root/.cache/pip

%files
nahrbtnik.py /app/nahrbtnik.py
soll.csv     /app/soll.csv

%workingdirectory /app

%environment
export LC_ALL=C
export LANG=C

%runscript
exec python3 "$@"

```

- Na podlagi definicijske datoteke zgradi SIF datoteko:

```
singularity build nahrbtnik.sif nahrbtnik.def
```

- Preizkusi vsebnik lokalno:

```
singularity exec nahrbtnik.sif python3 nahrbtnik.py
```

- Prenesi SIF datoteko na HPC sistem.
- Zaženi MPI program znotraj vsebnika:

```
mpirun -np 4 singularity exec nahrbtnik.sif python3
nahrbtnik.py
```

#### Opombe:

- Če MPI Python program vsebuje dodatne pakete, jih dodamo v `post` bloku kode.
- Če so potrebne dodatne datoteke, jih dodamo v `files` ter prilagodimo `%workingdirectory`.
- Zgornji postopek nam zagotavlja, da se MPI program izvaja znotraj vsebnika in da je okolje enotno na vseh vozliščih HPC sistema.

Primer izpisa `out` in `err` datoteke po izvedbi programa v Singularity je viden na slikah 3 in 4.

```

/nahrbtnik_16732340.out -
Job started
Host workdir: nahrbtnik_csv
Using container: nahrbtnik.sif
Mojster starta z 4 procesi...
Število predmetov: 3, maksimalna teža: 1000
Najvecja cena = 280
Nahrbtnik (originalni vrstni red): [1, 1, 1]

```

Slika 3: Izpis .out datoteke.

```

/nahrbtnik_16732340.err -

```

Slika 4: Izpis .err datoteke.

## 4 Zaključek

V nalogi smo obravnavali problem 0/1 nahrbtnika in ga reševali z metodo razveji in omeji. Algoritem smo implementirali v paralelnem okolju z uporabo MPI ter ga preizkusili v lokalnem in superračunalniškem okolju, kjer smo spoznali osnovne principe dela z okoljem HPC in sistemom za razdeljevanje opravil. Za zagotovitev prenosljivosti in ponovljivosti izvajanja smo algoritem zapakirali v vsebnik Singularity, kar omogoča enostavno izvajanje programa na različnih računalniških sistemih. Pridobljeno znanje bi bilo mogoče nadgraditi z nadaljnjo optimizacijo algoritma, zlasti z zmanjšanjem prostorske zahtevnosti, kar bi omogočilo obravnavo večjih množic predmetov v okolju HPC, kjer so pomnilniški viri omejeni.

## Literatura

- [1] M. Molnar, E. Smolič, M. Galun, D. Bokal. *Navigacija po svetu superračunalništva: Praktični vodnik za prijavo projekta, dostop in uporabo gručnega okolja*. Dianoia 9 (2025), 153–165.
- [2] Vegi Kalamar A. (2018), *Paralelni razveji in omeji algoritem BiqMac Solver*, magistrsko delo, Univerza v Mariboru, Fakulteta za naravoslovje in matematiko, Maribor.  
Pridobljeno iz: <https://dk.um.si/Dokument.php?id=129953&lang=slv>, dostopano: 3. 11. 2025
- [3] Wikipedia contributors, *Knapsack problem*, Wikipedia.  
Pridobljeno iz: [https://en.wikipedia.org/wiki/Knapsack\\_problem#0-1\\_knapsack\\_problem](https://en.wikipedia.org/wiki/Knapsack_problem#0-1_knapsack_problem), dostopano: 3. 11. 2025
- [4] *Singularity dokumentacija za uporabnike*.  
Pridobljeno iz: <https://docs.sylabs.io/guides/3.5/user-guide/introduction.html>, dostopano: 8. 12. 2025

- [5] *Open MPI dokumentacija za uporabnike.*  
Pridobljeno iz: <https://www.open-mpi.org/doc/>, dostopano: 3. 11. 2025
- [6] *HPC Vega - IZUM, Maribor, Slovenia.*  
Pridobljeno iz: <https://en-vegadocs.vega.izum.si/introduction/>, dostopano: 3. 11. 2025



# Hibridno modeliranje hrapavosti 3D tiskanih kovin

## Hybrid modelling roughness of 3D printing metal

Matej Babič<sup>1</sup>, Roman Šturm<sup>2</sup>

<sup>1</sup>Fakulteta za informacijske študije, Novo mesto, Slovenija

<sup>2</sup>Fakulteta za strojništvo, Univerza v Ljubljani, Ljubljana, Slovenija

---

### Povzetek

Selektivno lasersko taljenje (SLM), najpogosteje uporabljena tehnika za aditivno tiskanje kovinskih komponent, je učinkovita pri izdelavi funkcionalnih modelov in prototipov. Izhodni material je fin kovinski prah, običajno iz aluminijeve zlitine, nerjavnega oziroma orodnega jekla. Ta se tali plast za plastjo z zelo natančnim laserjem. Eksperimentalno delo je potekalo s 3D-tiskalnikom EOS M 290 in kovinskim prahom EOS Maraging Steel MS1. Ta študija ponuja svež pogled na hrapavost 3D tiskanja kovin in ima pomembne posledice za nadzor kakovosti v aditivni proizvodnji, kar lahko izboljša natančnost in učinkovitost 3D-tiskanja kovin. Cilj raziskave je razviti hibridni inteligentni sistem za opis hrapavosti 3D SLM kovinskih materialov.

*Ključne besede: SLM, aditivna tehnologija, EOS M 290, hibridno modeliranje, SVM, k-NN,*

### Abstract

Selective laser melting (SLM), the most widely used technique for additive printing of metal components, is effective in producing functional models and prototypes. The starting material is a fine metal powder, usually aluminum alloy, stainless or tool steel. This is melted layer by layer by a high-precision laser. The experimental work was carried out with the EOS M 290 3D printer and the EOS Maraging Steel MS1 metal powder. This study provides a fresh perspective on the roughness of 3D metal printing and has important implications for quality control in additive manufacturing, which can improve the accuracy and efficiency of 3D metal printing. The research aims to develop a hybrid intelligent system for describing the roughness of 3D SLM metal materials.

*Key words: SLM, additive technology, EOS M 290, hybrid modeling, SVM, k-NN*

---

## 1 UVOD

V zadnjih letih so 3D-tiskalniki [1] po vsem svetu pritegnili veliko pozornosti kot tehnologija, ki bo imela velik vpliv na proizvodnjo. 3D-tiskalnik je tehnologija, ki uporablja računalnik za izračun oblike na podlagi tridimenzionalnih (3D) CAD podatkov in nato na podlagi rezultatov teh izračunov nanaša materiale v plasteh, da ustvari tridimenzionalni objekt. Izraz 3D-tiskalnik se pogosto uporablja, ker ga je intuitivno lažje razumeti v primerjavi z dvodimenzionalnim (2D) tiskalnikom, ki tiska na navaden papir, vendar je mednarodno uradni izraz tehnologija aditivne proizvodnje.

Kovinski 3D-tiskalniki omogočajo integrirano proizvodnjo kompleksnih delov, ki bi jih bilo težko izdelati z običajnimi metodami rezanja, varjenja ali litja, kar omogoča izdelavo inovativnih strukturnih komponent z dodanimi novimi funkcijami. Na primer, deli, optimizirani za strukturno zasnovo z uporabo topološke optimizacije, imajo potencial, da dosežejo bistveno manjšo težo kot običajne zasnove s popolnoma drugačnimi strukturnimi slogi. Rešetkaste (mrežaste) strukture, ki združujejo kovine v mrežastem vzorcu, so edinstvene za kovinske 3D-tiskalnike.

Aditivna proizvodnja (AM) [2] je nova metoda izdelave izdelkov na osnovi 3D-tiskalnikov, ki ustvarja oblike z nanosom materialov v plasteh, in ima v primerjavi s

tradicionalnimi tehnikami množične proizvodnje tako prednosti kot slabosti. Trenutno ima množična proizvodnja veliko prednost v smislu produktivnosti, medtem ko aditivna proizvodnja blesti pri ustvarjanju unikatnih izdelkov in kompleksnih oblik.

Tehnologija aditivne proizvodnje se je začela v osemdesetih letih prejšnjega stoletja za uporabo pri izdelavi prototipov na osnovi smol, v zadnjih letih pa se je razširila na proizvodnjo visokozmogljivih delov z uporabo kovinskih prahov in keramike. Vse bolj se uporablja v letalski, vesoljski in avtomobilski industriji ter zdaj pritegne pozornost kot tehnologija za množično proizvodnjo v polnem obsegu.

Največja razlika med aditivno in subtraktivno proizvodnjo je v načinu ravnanja z materialom. Medtem ko subtraktivna proizvodnja vključuje odrezavanje nepotrebnih delov iz obdelovanega materiala za ustvarjanje oblike, aditivna proizvodnja vključuje izdelavo le zahtevane oblike. Ker je mogoče material graditi učinkovito, je glavna prednost aditivne proizvodnje majhna količina odpadnega materiala. Aditivna proizvodnja omogoča tudi oblikovanje kompleksnih notranjih struktur in votlih struktur za zmanjšanje teže v enem kosu. Zmožnost obdelave kompleksnih oblik prispeva k večji svobodi oblikovanja. Vendar pa je včasih lahko slabša od postopkov rezanja glede površinske obdelave in dimenzijske natančnosti, zato se v mnogih primerih predvideva, da se bo naknadna obdelava uporabljala v kombinaciji.

Tehnologija 3D-tiskanja je bila prvotno znana kot »hitro prototipiranje«. Kot že ime pove, je bila zelo cenjena kot tehnologija, ki omogoča hitro izdelavo prototipov in je bila široko sprejeta v proizvodnji. Takrat so bili 3D-tiskalniki tehnološko nezreli, niso mogli uporabljati visokokakovostnih materialov in so imeli omejeno natančnost, zato so veljali za tehnologijo, omejeno na aplikacije za izdelavo prototipov. Vendar pa se v zadnjih letih vse bolj pojavlja trend uporabe 3D-tiskanja ne le za izdelavo prototipov, temveč tudi za proizvodnjo. To je v veliki meri posledica širjenja novega koncepta, znanega kot oblikovanje za aditivno proizvodnjo, ki vključuje 3D-tiskanje že od načrtovanja in oblikovanja izdelkov. Ta oblikovalski pristop omogoča razvoj izdelkov, ki bi jih z uporabo običajnih proizvodnih tehnik težko izdelali. Proizvodni primeri izdelkov na osnovi 3D-tiskanja se po vsem svetu hitro povečujejo.

Potencial aditivne proizvodnje se pogosto obravnava v smislu omogočanja obdelave oblik, ki jih je bilo prej težko ustvariti, kar vodi do zmanjšanja stroškov z optimizacijo oblik. Vendar AM ni zgolj orodje za izboljšanje obstoječih proizvodnih procesov; gre za tehnologijo, ki bo temeljito preoblikovala celoten proces razvoja, proizvodnje in distribucije izdelkov ter povzročila revolucijo v proizvodnji.

Zahvaljujoč tehnologiji, znani kot strojno učenje, lahko računalnik skenira ogromne količine podatkov, najde skrite vzorce v podatkih in ustvari pravila za analizo neznanih podatkov [3]. V zadnjih letih je postala bolj znana kot tehnologija, ki je pozicionirana kot del "umetne inteligence". Ta tehnologija trenutno začne vplivati na številna področja, vključno z biologijo, avtonomno vožnjo, obdelavo slik in finančnim inženiringom. Sistem, znan kot strojno učenje, vključuje vnašanje ogromnih količin podatkov v računalnik in njihovo analizo z uporabo različnih algoritmov. Iskanje skritih značilnosti in vzorcev v podatkih je mogoče doseči z večkratnim dajanjem navodil računalniku. Ključnega pomena je pripraviti algoritme,

ki so primerni za vsak namen, saj se tukaj uporabljeni algoritmi spreminjajo glede na lastnosti in namen podatkov, ki se vnašajo v računalnik.

Cilj raziskave je razviti hibridni inteligentni sistem strojnega učenja za opis hrapavosti 3D tiskanih kovinskih materialov.

## 2 PRIPRAVA MATERIALA, EKSPERIMENTALNO DELO IN MODELIRANJE

Za 3D tiskanje kovinskih vzorcev smo uporabili napravo EOS M 290 (slika 1). Tehnologija neposrednega navarjanja se izvaja v zaprti komori, običajno v atmosferi dušika, kjer med delovnim procesom usmerjeni laserski žarek deluje na zgornjo plast prahu in na njem tvori talino, kar zagotavlja povezavo z izvornim materialom. Povprečna debelina ene plasti prahu je 0,1 mm. Za večje velikosti izdelkov se postopek nanašanja prahu ponovi. Ponovi pa se tudi lasersko taljenje prahu na izbranih pozicijah. Enakomerno debelino plasti zagotavlja nalagalno rezilo.



Slika 1: EOS M 290 (Sieva d.o.o., 3D LAB)

Za preizkuse smo uporabili material EOS Maraging Steel, kjer je prah orodnega jekla namenjen za obdelavo na sistemih EOS DMLS. Po kemični sestavi prah ustreza ameriški klasifikaciji 18 % Ni Maraging 300, evropski 1.2709 in nemški X3NiCoMoTi 18-9-5. Za to vrsto jekla so značilne zelo dobre mehanske lastnosti, ki jih lahko izboljšamo z naknadno toplotno obdelavo izločevalnega utrjevanja. Deli, izdelani iz EOS Maraging Steel MS1, so po SLM postopku izdelave dosegli trdoto okoli 30 HRC, po izločevalnem utrjevanju pa več kot 50 HRC (staranjem pri 490 °C (914 °F) 6 ur). Tako v izdelanem kot v staranem stanju je mogoče dele strojno obdelovati, erodirati, variti, mikrokrokovati, polirati in po potrebi površinsko zaščititi. Zaradi metode plastne gradnje imajo deli določeno anizotropijo, ki jo je mogoče zmanjšati ali odstraniti z ustrezno toplotno obdelavo – npr. z žarjenjem pri 940 °C (1724 °F) 2 uri. Izdelali smo 17 kockastih vzorcev s stranico 10 mm pri različnih izdelovalnih parametrih, prikazanih v tabeli 1. Na površini vzorcev smo potem merili hrapavost. Hrapavost površine se nanaša na celoto nepravilnosti, ki tvorijo njen relief. Za merjenje hrapavosti površine selektivno laserskega taljenja (SLM) [4] 3D-tiskanih materialov uporabljamo profilometer. Kontaktna profilometrija je kvantitativna tehnika, za katero je znano, da odraža nepravilnosti površinskega

profila 3D SLM kovinskih materialov. Najpogostejši parameter, izračunan iz profila hrapavosti, je povprečna hrapavost ( $R_a$ ).

Za modeliranje hrapavosti smo uporabili metode inteligentnih sistemov. Uporabili smo dve metodi strojnega učenja, in sicer metodo k-najbližjih sosedov (k-NN) in metodo podpornih vektorjev (SVM).

Algoritem k-najbližjih sosedov (k-NN) [5] je nelinearni nadzorovani učni klasifikator, ki uporablja bližino za klasifikacije ali napovedi o skupinah posameznih podatkovnih točk. Je eden najbolj priljubljenih in najpreprostejših klasifikacijskih in regresijskih klasifikatorjev, ki se danes uporabljajo v strojnem učenju. Algoritem kNN se lahko uporablja tako za regresijske kot klasifikacijske naloge, vendar se običajno uporablja kot klasifikacijski algoritem, ki deluje na predpostavki, da so podobne točke blizu druga drugi. Pri problemih klasifikacije se oznake razredov dodelijo na podlagi večinskega glasu, tj. uporablja se oznaka, ki je najpogosteje zastopana okoli določene podatkovne točke. Regresijski problemi uporabljajo podobne koncepte kot problemi klasifikacije, vendar se v tem primeru vzame povprečje k najbližjih sosedov in naredi napoved o klasifikaciji. Glavna razlika je v tem, da se klasifikacija uporablja za diskretne vrednosti, regresija pa za zvezne vrednosti. Preden pa se lahko izvede klasifikacija, pa je treba definirati razdaljo. Najpogosteje uporabljena je evklidska razdalja, o kateri bomo podrobneje razpravljali kasneje. Za to raziskavo so bili uporabljeni naslednji parametri k-NN: število sosedov 8, evklidska metrika in enotna teža.

Metoda podpornih vektorjev (SVM) [6] je eden najbolj znanih algoritmov v strojnem učenju. V današnjem svetu, kjer se tehnologija umetne inteligence hitro razvija in širi, so SVM uporabni za aplikacije, kot so zelo natančno napovedovanje, zaznavanje in identifikacija, ter bodo prispevali tudi k izboljšanju delovne učinkovitosti in produktivnosti. »Podporni vektor« v SVM se nanaša na podatke, ki so najbližje najboljši premici ujemanja, ki deli podatke. Podporni vektorji igrajo pomembno vlogo pri nalogah klasifikacije z SVM. Določanje podpornega vektorja omogoča razjasnitev podatkov, ki bodo služili kot osnova za delilne črte itd. Ko je osnova določena, se napove klasifikacija razreda glede na to, na kateri strani so ciljni podatki. Eden od ciljev SVM je maksimirati razdaljo med podpornim vektorjem in ciljnim podatki, s čimer se izboljša natančnost ocene klasifikacije. Predvsem binarna klasifikacija z uporabo dveh možnosti je eno od področij, na katerih blestijo SVM, poleg tega pa lahko obdelujejo tudi večvrednostno klasifikacijo, ki uporablja več binarnih klasifikacijskih algoritmov. Čeprav so manj pogosti kot klasifikacija, jih je mogoče uporabiti tudi za regresijo (napovedovanje vrednosti). V tej študiji so bili uporabljeni naslednji parametri SVM:

- SVM, stroški (C): 1,00;
- epsilon izguba regresije: 0,1;
- Jedro: RBF;
- Optimizacijski parametri: numerična toleranca: 0,001;
- iteracijska omejitev: 100.

Hibridno strojno učenje (HML) [7] združuje različne modele, algoritme ali pristope strojnega učenja (kot je tradicionalno strojno učenje z globokim učenjem ali več enostavnejših

modelov), da bi izkoristilo njihove skupne prednosti in premagalo omejitve posameznih metod za robustnejše, natančnejše in razumljivejše rešitve, zlasti pri kompleksnih in raznolikih podatkih. Združuje strukturirano znanje (fiziko, pravila) z učenjem, ki ga poganjajo podatki, ter izboljšuje učinkovitost nalog, od analize slik do napovednega vzdrževanja, z uravnoteženjem učinkovitosti, natančnosti in razumljivosti. V tej raziskavi bomo uporabili hibridno strojno učenje, povprečje dve metod SVM in k-NN. Torej končna napoved hibridnega strojnega učenja bo povprečna vrednost napovedi SVM in k-NN.  $H = (SVM + k-NN)/2$ .

### 3 REZULTATI IN RAZPRAVA

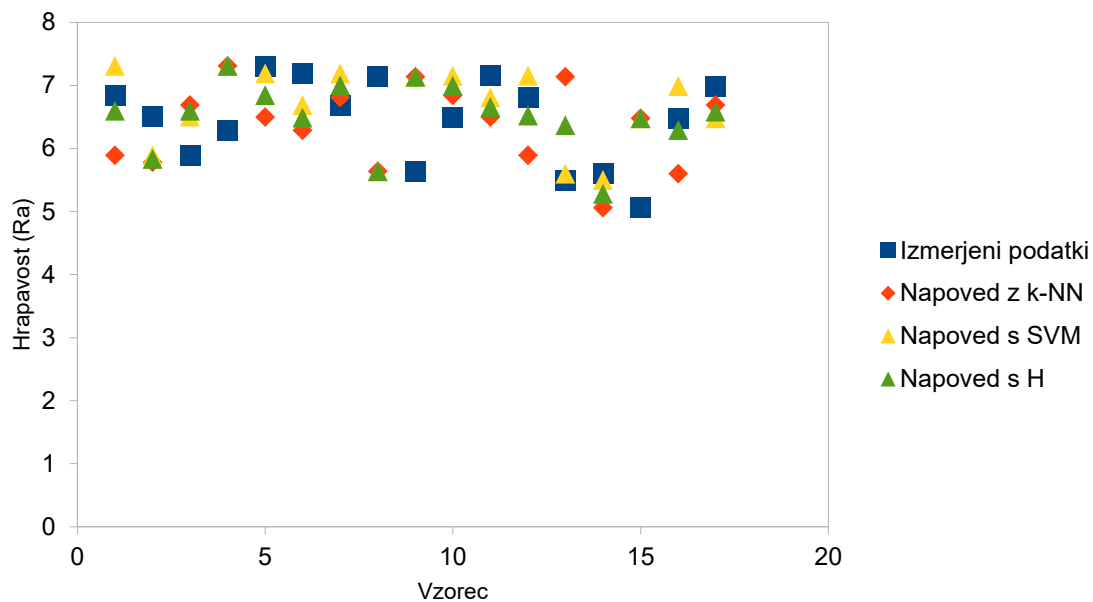
Tabela 1 prikazuje parametre izdelave vzorcev SLM in izmerjeno hrapavost Ra. V prvem stolpcu smo označili vzorce od S1 do S17. Drugi stolpec predstavlja moč laserja v W, označeno kot X1. Tretji stolpec ponazarja hitrost laserja v mm/s, označeno kot X2. Zadnji stolpec predstavlja hrapavost Ra ( $\mu\text{m}$ ). Tabela 2 prikazuje napovedne vrednosti hrapavosti z metodami k-NN, SVM in HML. K-NN model nam da natančnost 91,8 %, SVM model nam da natančnost 95,9 % in H model nam da 99,4 % natančnost. Slika 2 prikazuje izmerjene in prediktirane podatke za hrapavost Ra. Slika 3 prikazuje razmerje med izmerjenimi in prediktiranimi podatki za hrapavost Ra.

Vzorec	Moč (W) X1	Hitrost (mm/s) X2	Hrapavost Ra ( $\mu\text{m}$ ) Y
S1	320	1000	6,84
S2	320	1150	6,51
S3	320	1300	5,89
S4	270	850	6,29
S5	270	1000	7,31
S6	270	1150	7,19
S7	270	1300	6,69
S8	220	700	7,14
S9	220	850	5,64
S10	220	1000	6,50
S11	220	1150	7,15
S12	220	1300	6,81
S13	170	700	5,50
S14	170	850	5,60
S15	170	1000	5,06
S16	170	1150	6,48
S17	170	1300	6,99

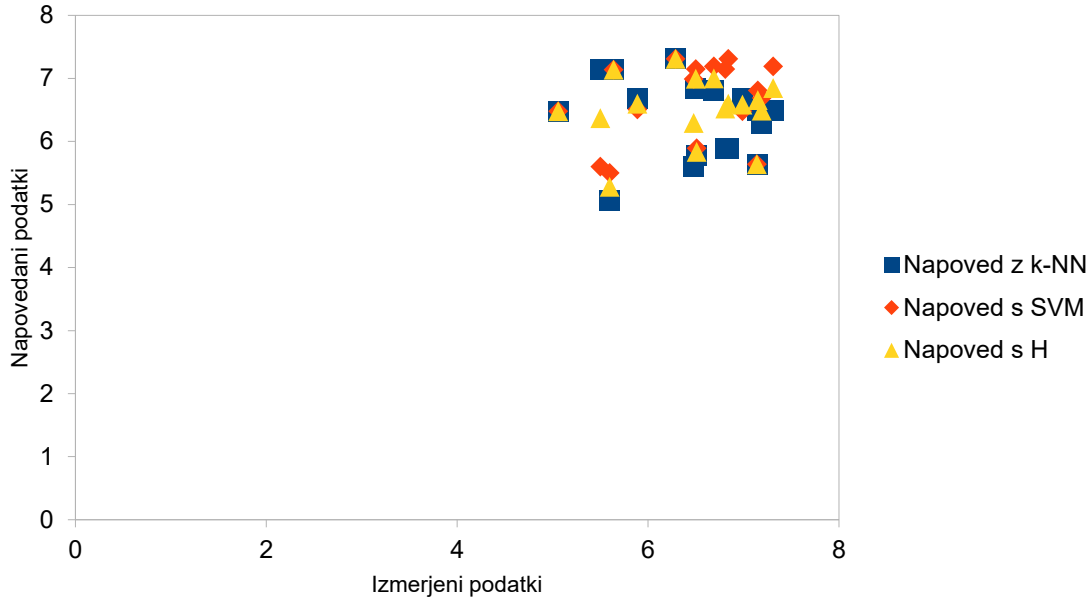
Tabela 1: Parameteri 3D tiskalnika in izmerjena hrapavost

Vzorec	Napoved z k-NN	Napoved s SVM	Napoved s HML
S1	5,89	7,31	6,60
S2	5,78	5,89	5,83
S3	6,69	6,51	6,60
S4	7,31	7,31	7,31
S5	6,5	7,19	6,84
S6	6,29	6,69	6,49
S7	6,81	7,19	7,00
S8	5,64	5,64	5,64
S9	7,14	7,14	7,14
S10	6,84	7,15	6,99
S11	6,5	6,81	6,65
S12	5,89	7,15	6,52
S13	7,14	5,60	6,37
S14	5,06	5,50	5,28
S15	6,48	6,48	6,48
S16	5,6	6,99	6,29
S17	6,69	6,48	6,58

Tabela 2: Napovedne vrednosti hrapavosti z metodami k-NN, SVM in H



Slika 2: Izmerjeni in prediktirani podatki za hrapavost Ra



Slika 3: Razmerje med izmerjenimi in prediktiranimi podatki za hrapavost Ra

Postopek učenja in validacije je bil izveden na naslednji način. Osemdeset odstotkov nabora podatkov je bilo uporabljenih za učenje modela, preostalih dvajset odstotkov pa je bilo namenjenih testiranju. Učinkovitost predlaganega modela strojnega učenja je bila ocenjena s tehniko k-kratne navzkrižne validacije [8]. Za ugotovitve 2-kratne navzkrižne validacije je bil izbran  $k = 2$ . Nabor podatkov je bil nato z naključnim premeščanjem razdeljen na dva enako velika nabora,  $d_0$  in  $d_1$ . Ta tehnika se običajno uporablja za razdelitev podatkovnega niza na polovico po premešanju. Model je bil nato usposobljen na  $d_0$  in validiran na  $d_1$ , nato pa usposobljen na  $d_1$  in validiran na naborih  $d_0$ . Za prilagoditev hiperparametrov je bil uporabljen pristop iskanja rid [9].

Porazdelitev napovedi se lahko znatno razlikuje, tudi če je natančna glede povprečnih vrednosti in variabilnosti. Zato je treba pogledati tudi linearne korelacije med izmerjenimi in pričakovanimi vrednostmi. Tabela 3 prikazuje Pearsonovo korelacijo med izmerjenimi rezultati in koeficientom napovedi. Razdalje med izmerjenimi in pričakovanimi vrednostmi so bile uporabljene za primerjavo, izračunano v smislu odstotnih napak glede na povprečne vrednosti hrapavosti. Ker je povprečna kvadratna napaka (MSE) bistveno orodje za ocenjevanje natančnosti napovedi v statističnem modeliranju, je bila vključena tudi v tabelo 3. Tudi koren povprečne kvadratne napake (RMSE) ali koren povprečne kvadratne napake (RMSE) je ena od dveh tesno povezanih in pogosto uporabljenih mer razlik med resničnimi ali napovedanimi vrednostmi na eni strani in opazovanimi vrednostmi ali ocenjevalnimi na drugi strani, ki je predstavljen tudi v tabeli 3.

Slika 3 prikazuje grafični prikaz podatkov z enakimi Pearsonovimi korelacijskimi koeficienti za izmerjeno in napovedano hrapavost. Odvisno od posameznih metod strojnega učenja so lahko na primer taki koeficienti  $-0,1741$  ali  $0,3122$  za vrednosti trdote, napovedane s k-NN, SVM in HML. Zdi se, da sta obe tehniki napovedovanja enako učinkoviti glede na razmerje med koeficienti.

	k-NN	SVM	HML
Error (distanca)	8,2%	4,1%	0,6%
Pearson koeficient	-0,1741	0,3122	0,0814
MSE	0,90793	0,56971	0,61253
RMSE	0,95286	0,75479	0,78265

Tabela 3: Korelacija med izmerjenimi rezultati in napovedmi

Netočnosti glede razlik med izmerjenimi in pričakovanimi vrednostmi so prikazane v drugi vrstici tabele 3. Izračunane so kot odstotne napake glede na povprečne vrednosti trdote. V večini primerov je jasno, da je bil vsak algoritem sposoben ustvariti natančne napovedi. Natančnost napovedi vsake metode je bila določena tako, da se je izmerjena vrednost delila z absolutno razliko med napovedanimi in izmerjenimi vrednostmi. Nato je bilo določeno povprečje vseh podatkov. Natančnost napovedi je bila tako 91 % za model k-NN, 95,9 % za model SVM, in 99,4 % za hibridni model.

Srednja kvadratna napaka (MSE), ki daje jasno oceno delovanja modela, je prikazana v tretji vrstici tabele 3. Za njen izračun je bila uporabljena povprečna kvadratna razlika med pričakovanimi in dejanskimi številkami. Ker je MSE kvadratna, se je izkazala še posebej uporabno za iskanje modelov, ki se izogibajo pomembnim napakam pri napovedovanju. Ker so modeli SVM pokazali nižjo vrednost MSE kot druga modela, lahko njihovo delovanje štejeemo za boljše.

Četrta vrstica prikazuje koren srednjega kvadratnega odklona (RMSE) ali korensko srednjo kvadratno napako (RMSE). Najboljši rezultat ima SVM model 0,75479.

## 5 ZAKLJUČEK

V tem članku smo predstavili modeliranje hrapavosti površine vzorcev, izdelanih s SLM postopkom. Kljub temu 3D-tiskanje kovin ponuja številne prednosti in postaja vse bolj priljubljeno v industriji, zlasti glede svobode gradnje. Ker je komponenta izdelana plast za plastjo v željeni obliki, je mogoče uporabiti tudi težko obdelovalne materiale. S tem postopkom aditivne proizvodnje je mogoče izdelati celo zelo zapletene vzorce in gibljive dele. Poleg tega je proizvodnja brez orodja zelo hitra in zanesljiva. Ekonomsko učinkovitost je mogoče še povečati z uporabo čim manj odpadkov, zlasti pri tiskanju kovin. Dandanes je mogoče zaradi industrijskega 3D-tiskanja komponente tiskati z natančnostjo 0,1 mm. Komponente te generativne proizvodne tehnike izpolnjujejo tudi najzahtevnejše specifikacije. Komponente, ki bi bile prej predrage ali jih morda ne bi bilo mogoče izdelati s konvencionalnimi proizvodnimi metodami, je zdaj mogoče izdelati, kar je stroškovno zelo ugodno tudi pri majhnih količinah. 3D-tiskanje kovin je koristno ne le za tradicionalne kupce visokotehnoloških komponent, kot sta avtomobilska ali letalska industrija, temveč tudi za področja, kot so strojništvo, orodjarna in medicina. Za napoved poroznosti SLM vzorcev smo uporabili k-NN, SVM in hibridni model. Študija ponuja svež pogled na modeliranje-določanje hrapavosti in ima pomembne učinke za nadzor kakovosti v aditivni proizvodnji, kar lahko izboljša natančnost in učinkovitost 3D-tiskanja kovin. Namen študije je modelirati hrapavost 3D-SLM kovinskih materialov z uporabo inteligentnega systemskega pristopa. Za napoved poroznosti SLM vzorcev smo uporabili k-NN, SVM in HML hibridni model. Ugotovili smo, da najbolj natančne rezultate predikcije da SVM model.

## Literatura

- [1] Shahrubudin, N.; Lee, T.C.; Ramlan, R. An overview on 3D printing technology: Technological, materials, and applications. *Procedia Manuf.* 2019, 35, 1286–1296. <https://doi.org/10.1016/j.promfg.2019.06.089>.
- [2] Groth, J.-H.; Magnini, M.; Tuck, C.; Clare, A. Stochastic design for additive manufacture of true biomimetic populations. *Addit. Manuf.* 2022, 55, 102739. <https://doi.org/10.1016/j.addma.2022.102739>.
- [3] Gao, H.; Kou, G.; Liang, H.; Zhang, H.; Chao, X.; Li, C.C.; Dong, Y. Machine learning in business and finance: A literature review and research opportunities. *Financ. Innov.* 2024, 10, 86. <https://doi.org/10.1186/s40854-024-00629-z>.
- [4] Soni, N.; Renna, G.; Leo, P. Advancements in Metal Processing Additive Technologies: Selective Laser Melting (SLM). *Metals* 2024, 14, 1081. <https://doi.org/10.3390/met14091081>.
- [5] Halder, R.K.; Uddin, M.N.; Uddin, M.A.; Aryal, S.; Khraisat, A. Enhancing K-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications. *J. Big Data* 2024, 11, 113.
- [6] Abe, S. *Support Vector Machines for Pattern Classification*; Springer: London, UK, 2005.
- [7] Jorge Paredes, Danilo Chávez, Ramiro Isa-Jara, Diego Vargas. A hybrid machine learning algorithm approach to predictive maintenance tasks: A comparison with machine learning algorithms, *Results in Engineering*, Volume 26, 2025, 105137, ISSN 2590-1230, <https://doi.org/10.1016/j.rineng.2025.105137>.
- [8] Anandan, B.; Manikandan, M. Machine learning approach with various regression models for predicting the ultimate tensile strength of the friction stir welded AA 2050-T8 joints by the K-Fold cross-validation method. *Mater. Today Commun.* 2023, 34, 105286.
- [9] Ogunsanya, M.; Isichei, J.; Desai, S. Grid search hyperparameter tuning in additive manufacturing processes. *Manuf. Lett.* 2023, 35 (Supplement), 1031–1042.



## VABILO AVTORJEM

Dianoia (grško δίανοια) po Platonu označuje védenje, razmišljanje o modelih stvarnosti, o naravoslovno-matematičnih in tehničnih temah. Uporabljajo ga matematiki (modeliranje) in znanstveniki (formuliranje problema), inženirji (načrtovanje sistema). Opredeljuje kompetenco, proces ali rezultat diskurzivnega razmišljanja, za razliko od neposrednega razumevanja obravnavane tematike. Aristotel to védenje naprej razdeli na teoretično (episteme) in praktično (phronesis).

Dianoia po Platonu torej označuje vmesni nivo človeškega spoznanja, prehod od intuitivnih občutkov do najglobljega spoznanja dejanskosti. Tako je idealna oznaka za objave v pričujoči reviji, ki povezujejo teoretična, znanstvena izhodišča z njihovo uporabno namembnostjo. Študentje, avtorji teh člankov, ste na prehodu od učenja k delu, od teoretičnega h konkretnemu, ki vas bo pripeljalo do kruha, do dela, s katerim boste odigrali svojo vlogo v družbi. Na tem prehodu pa poleg znanja, ki ga ponuja redno izobraževanje, potrebujete tudi izkušnje s konkretnih izzivov in mehke kompetence sodelovanja v ekipah delodajalcev, k čemur vas spodbuja in vam pri tem pomaga revija Dianoia.

V reviji bomo objavljali poljudne in strokovne članke s področja naravoslovja, matematike ali znanosti, ki uporabljajo znanja teh področij. Ciljna publika bralcev so v prvi vrsti delodajalci, ki tovrstna znanja potrebujejo in želijo izvedeti, kaj je kdo zanimivega razmislil na njihovem področju. V drugi vrsti so ciljna publika študentje, ki iščejo zamisli za svojo poklicno pot in lahko v reviji najdejo navdih za lastna raziskovanja in iskanje stikov s trgom dela.

Za kakovost izdelkov bo skrbel uredniški odbor in uredniški svet, v katerih so vrhunski strokovnjaki, povezani s področji, ki jih revija obravnava. Članki bodo anonimno recenzirani, o objavi pa na podlagi recenzije odloča uredniški odbor. Priporočljivo je, da avtorji besedilo spremenijo v skladu s priporočili recenzentov in da popravljeni članek z utemeljitvijo sprejema ali zavrnitve sprememb ponovno pošljejo v pregled. Uredništvo lahko objavo članka zavrne, če vsebinsko ali po merilih kakovosti ne ustreza standardom revije, o čemer avtorje obvestimo v najkrajšem možnem času.

S prispevkom v reviji bodo avtorji spodbujali širjenje znanja s področja naravoslovja in matematike ter tehnike oziroma izobraževanja teh področij in svoje poglede prenašali na trg dela in na prihajajoče generacije.

## NAVODILA AVTORJEM

Avtorje prosimo, da pri pripravi članka upoštevajo naslednja navodila.

Če je članek napisan v slovenščini, naj ima angleški prevod naslova, povzetka in ključnih besed. Veseli bomo tudi prispevkov v angleščini, ki pa morajo imeti naslov, razširjen povzetek v obsegu 300 – 400 besed in ključne besede v slovenščini. Ključnih besed naj bo do šest.

Prispevki naj bodo zanimivi za širši krog bralcev. Ključna je intuitivna predstavitev zamisli in rezultatov, podrobnosti pa lahko ostanejo prihranjene za morebitni znanstveni članek, ki bi bil nadgradnja članka, objavljenega v reviji Dianoia.

Članek naj vsebuje naslov, ime avtorja (avtorjev) in sedež ustanove, kjer avtor(ji) dela(jo). Sledi naj povzetek, z največ 150 besedami, seznam ključnih besed in besedilo, ki ne presega 3000 besed. Besedilo naj bo zapisano v urejevalniku besedil MS Word 2010 oz. kasnejši ali LaTeX in naj uporablja objavljeno predlogo. Slike in tabele morajo biti oštevilčene in imeti natančen opis, da jih lahko razumemo brez preostalega besedila. Slike v elektronski obliki naj bodo visoke kakovosti v formatu PNG ali JPEG.

Prispevek v PDF obliki pošljite na naslov [dianoia@um.si](mailto:dianoia@um.si) z zadevo: »Za revijo Dianoia«. Če bo sprejet v objavo, vas bomo prosili za izvorno obliko prispevka.