

SVEUČILIŠTE JURJA DOBRILE U PULI  
FAKULTET INFORMATIKE

Neven Nižić

**”Optimizacija raspodjele projektnih aktivnosti primjenom genetskih  
algoritama i Monte Carlo simulacije”**

DIPLOMSKI RAD

Pula, rujan, 2025. godine

SVEUČILIŠTE JURJA DOBRILE U PULI  
FAKULTET INFORMATIKE

Neven Nižić

**”Optimizacija raspodjele projektnih aktivnosti primjenom genetskih  
algoritama i Monte Carlo simulacije”**

DIPLOMSKI RAD

**JMBAG: 0303118917, izvanredni student  
Studijski smjer: Informatika**

**Kolegij: Modeliranje i simulacije  
Znanstveno područje : Društvene znanosti  
Znanstveno polje : Informacijske i komunikacijske znanosti  
Znanstvena grana : Informacijski sustavi i informatologija**

**Mentor: doc.dr.sc. Darko Etinger**

Pula, rujan, 2025. godine

## Sažetak

Upravljanje projektima često uključuje složene odluke vezane uz raspodjelu aktivnosti i resursa, osobito u uvjetima nesigurnosti i vremenskih ograničenja. Tradicionalne metode kao što su PERT i CPM često ne uspijevaju obuhvatiti stohastičku prirodu stvarnih projekata. U ovom radu predlaže se model koji kombinira genetske algoritme i Monte Carlo simulaciju s ciljem postizanja robusne optimizacije raspodjele projektnih aktivnosti. Genetski algoritam koristi se za pretraživanje prostora mogućih rješenja, dok Monte Carlo simulacija omogućava procjenu utjecaja varijabilnih trajanja aktivnosti na ukupno trajanje projekta. Eksperimentalna evaluacija modela provodi se na simuliranim projektnim podacima, a rezultati pokazuju poboljšanu robusnost i učinkovitost u odnosu na determinističke pristupe.

**Ključne riječi :** projektno upravljanje, genetski algoritam, Monte Carlo simulacija, optimizacija rasporeda, raspodjela aktivnosti

## Abstract

Project management often involves complex decisions related to the allocation of activities and resources, especially under uncertainty and time constraints. Traditional methods such as PERT and CPM frequently fail to capture the stochastic nature of real-world projects. This thesis proposes a model that combines genetic algorithms and Monte Carlo simulation to achieve robust optimization of project activity allocation. The genetic algorithm is used to explore the space of possible solutions, while Monte Carlo simulation estimates the impact of variable activity durations on the overall project timeline. The model is experimentally evaluated using simulated project data, and results indicate improved robustness and efficiency compared to deterministic approaches.

**Keywords :** project management, genetic algorithm, Monte Carlo simulation, schedule optimization, activity allocation

# Sadržaj

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Uvod</b>   | <b>1</b>  |
| 1.1      | Uvodni dio . . . . .  | 1         |
| 1.2      | Motivacija . . . . .  | 1         |
| 1.3      | Rizici i nesigurnosti u projektnom upravljanju . . . . .            | 1         |
| 1.4      | Ciljevi i istraživačka pitanja . . . . .                            | 2         |
| <b>2</b> | <b>Teorijska podloga</b>  | <b>3</b>  |
| 2.1      | Knapsack problem . . . . .  | 3         |
| 2.2      | Genetski algoritmi . . . . .  | 3         |
| 2.3      | Monte Carlo simulacija . . . . .                                    | 4         |
| 2.4      | PERT metoda i trokutasta distribucija . . . . .                     | 5         |
| 2.5      | Zaključak . . . . .   | 5         |
| <b>3</b> | <b>Model problema</b>   | <b>7</b>  |
| 3.1      | Ograničenja . . . . .   | 7         |
| 3.2      | Cilj optimizacije . . . . .   | 7         |
| 3.3      | Grafički prikaz modela . . . . .                                    | 7         |
| 3.4      | Zaključak . . . . .   | 8         |
| <b>4</b> | <b>Implementacija</b>   | <b>9</b>  |
| 4.1      | Korištene biblioteke . . . . .                                      | 9         |
| 4.2      | Struktura sustava . . . . .   | 9         |
| 4.3      | Modeliranje nesigurnosti: Monte Carlo simulacija . . . . .          | 12        |
| 4.4      | Optimizacijski pristup: Genetski algoritam . . . . .                | 12        |
| 4.5      | Vizualizacija . . . . .   | 14        |
| <b>5</b> | <b>Eksperimenti i analiza rezultata</b>                             | <b>15</b> |
| 5.1      | Postavke okruženja i testni podaci . . . . .                        | 15        |
| 5.2      | Eksperimentalni dizajn . . . . .                                    | 15        |
| 5.3      | Eksperiment 1: Analiza parametara i kalibracija genetskog algoritma | 15        |
| <b>6</b> | <b>Zaključak</b>  | <b>18</b> |
|          | <b>Literatura</b>   | <b>19</b> |
|          | <b>Popis slika</b>  | <b>21</b> |
|          | <b>Popis tablica</b>  | <b>22</b> |

# 1 Uvod

## 1.1 Uvodni dio

Jedan od modela koji dobro opisuje izazove alokacije resursa je *problem ruksaka* (*Knapsack Problem*), gdje se ograničeni resursi moraju raspodijeliti na način koji maksimizira ukupnu korist [1]. U kontekstu projektnih aktivnosti, to znači odabrati skup zadataka koji će donijeti najveći povrat ulaganja (ROI) uz poštivanje vremenskih i resursnih ograničenja. Tradicionalne metode poput PERT-a (*Program Evaluation and Review Technique*) i CPM-a (*Critical Path Method*) [2, 3] pružaju okvir za planiranje i praćenje projekata, no često zanemaruju složene kombinacije zadataka i inherentnu neizvjesnost u procjenama trajanja i troškova.

Pregled literature pokazuje da se genski algoritmi (GA) uspješno primjenjuju na NP-teške optimizacijske probleme [4, 5], dok Monte Carlo simulacije (MC) omogućuju modeliranje i kvantifikaciju neizvjesnosti [6, 7]. Međutim, u literaturi postoji ograničen broj radova koji integriraju ove dvije metode u jedinstven hibridni pristup optimizaciji projektnih aktivnosti [8, 9]. Praznina se očituje u nedostatku sustavne analize kako kombinacija GA i MC može povećati robusnost rješenja u uvjetima visoke neizvjesnosti i složenih resursnih ograničenja.

## 1.2 Motivacija

Motivacija za ovaj rad proizlazi iz činjenice da se u stvarnim projektnim uvjetima planovi rijetko odvijaju točno onako kako su prvotno predviđeni [3]. Promjene u dostupnosti resursa, nepredviđeni rizici, promjene zahtjeva te pogrešne procjene trajanja zadataka česti su razlozi odstupanja od plana. Klasične metode planiranja projekata, iako korisne, često ne uzimaju u obzir dinamičke promjene i stohastičku prirodu projektnih parametara [10].

Upravo zbog toga potrebne su metode koje ne samo da optimiziraju raspodjelu resursa, već i uvažavaju nesigurnosti te omogućuju procjenu rizika povezanih s odabranim rješenjima. Genski algoritmi pružaju snažan alat za globalnu optimizaciju složenih problema pretrage [5], dok Monte Carlo simulacije omogućuju procjenu varijabilnosti i rizika rješenja [7]. Kombiniranjem ovih metoda moguće je razviti pristup koji generira kvalitetna rješenja, a pritom osigurava njihovu robusnost u uvjetima nesigurnosti [9].

## 1.3 Rizici i nesigurnosti u projektnom upravljanju

Rizici u projektnom upravljanju obuhvaćaju sve događaje ili uvjete koji, ako se pojave, mogu negativno utjecati na ciljeve projekta [11]. Oni mogu biti tehničke, organizacijske, financijske ili tržišne prirode, a često su međusobno povezani [10]. Nesigurnosti proizlaze iz nepotpunih informacija, promjenjivih uvjeta i nemogućnosti preciznog predviđanja budućih događaja [12]. Upravljanje rizicima uključuje njihovu identifikaciju, procjenu i razvoj strategija odgovora [11].

Monte Carlo simulacije predstavljaju jedan od najučinkovitijih pristupa kvantitativnoj procjeni rizika [13]. Kroz generiranje velikog broja scenarija temeljenih

na slučajnim varijacijama ulaznih parametara, moguće je procijeniti distribuciju mogućih ishoda i vjerojatnosti ostvarenja ciljeva projekta [7].

## 1.4 Ciljevi i istraživačka pitanja

Ovaj diplomski rad ima za cilj razviti i evaluirati hibridni GA-MC model za optimizaciju raspodjele projektnih aktivnosti. Doprinos rada je trojak:

- **Teorijski doprinos:** povezivanje optimizacijskih tehnika s metodama procjene neizvjesnosti u kontekstu projektnih aktivnosti.
- **Metodološki doprinos:** razvoj integriranog modela koji povezuje evolucijsku optimizaciju i statističku simulaciju.
- **Praktični doprinos:** demonstracija primjenjivosti pristupa na sintetičkim projektnim podacima i analiza performansi modela.

Istraživačka pitanja na koja ovaj rad želi odgovoriti su:

1. Može li hibridni GA-MC pristup postići bolje rezultate optimizacije u odnosu na samostalnu primjenu GA ili MC?
2. U kojoj mjeri Monte Carlo simulacija može poboljšati robusnost rješenja dobivenih genskim algoritmom?
3. Kakav je utjecaj kombiniranog pristupa na stabilnost i pouzdanost optimizacije u uvjetima neizvjesnosti?

Ciljevi istraživanja su:

- Dizajnirati i implementirati integrirani GA-MC model.
- Evaluirati performanse modela na različitim scenarijima.
- Usporediti rezultate hibridnog pristupa s rezultatima pojedinačnih metoda.

Sinergija između genskih algoritama i Monte Carlo simulacija proizlazi iz njihove komplementarne prirode: GA učinkovito pretražuje veliki prostor mogućih rješenja i pronalazi visokokvalitetne kandidate [5], dok MC kvantificira rizik i procjenjuje varijabilnost tih rješenja kroz stohastičko modeliranje [7]. Kombinacija ovih metoda rezultira robusnim pristupom koji ne samo da optimizira projektne aktivnosti, već i uzima u obzir realne nesigurnosti u planiranju.

Struktura rada organizirana je na sljedeći način:

- **Poglavlje 2** daje pregled relevantne literature i teorijskih osnova problema.
- **Poglavlje 3** opisuje korištenu metodologiju, uključujući principe rada GA i MC.
- **Poglavlje 4** prikazuje implementaciju hibridnog modela.
- **Poglavlje 5** opisuje plan i rezultate eksperimentalnog vrednovanja modela.
- **Poglavlje 6** sadrži zaključak i preporuke za budući rad.

## 2 Teorijska podloga

Ovo poglavlje pruža pregled temeljnih teorijskih koncepata ključnih za razumijevanje predloženog modela optimizacije projektnih aktivnosti. Detaljno će se objasniti Problem ruksaka kao osnova za formulaciju problema raspodjele resursa, Genetski algoritmi kao optimizacijska metaheuristika, te Monte Carlo simulacija i PERT metoda kao alati za modeliranje i analizu nesigurnosti u projektnom upravljanju.

### 2.1 Knapsack problem

Problem ruksaka (engl. *Knapsack Problem*) jedan je od najpoznatijih i najčešće proučavanih problema kombinatorne optimizacije, svrstan u klasu NP-teških problema [5]. U osnovnoj verziji, cilj je odabrati podskup objekata s pridruženom težinom i vrijednošću, s ciljem maksimiziranja ukupne vrijednosti odabranih objekata, pri čemu njihova ukupna težina ne smije prelaziti zadani kapacitet ruksaka.

Formalno, za skup od  $n$  objekata, gdje svaki objekt  $i$  ima težinu  $w_i$  i vrijednost  $v_i$ , te uz zadani kapacitet ruksaka  $W$ , cilj je maksimizirati funkciju:

$$\max \sum_{i=1}^n v_i x_i \quad \text{uz ograničenje} \quad \sum_{i=1}^n w_i x_i \leq W, \quad x_i \in \{0, 1\}$$

gdje je  $x_i = 1$  ako je objekt  $i$  odabran, a  $x_i = 0$  ako nije.

U kontekstu upravljanja projektima, ovaj se problem često pojavljuje u složenijim varijantama, poput višedimenzionalnog problema ruksaka (Multi-Dimensional Knapsack Problem – MDKP). U MDKP-u, osim jedne težine, svaki objekt (projektna aktivnost) ima više dimenzija "težine" koje predstavljaju različite vrste resursa (npr. vrijeme, budžet, broj radnika, specifična oprema). Kapacitet ruksaka tada predstavlja ograničenja za svaku od tih dimenzija. Projektne aktivnosti se mogu interpretirati kao objekti s određenim trajanjem, troškom i vrijednošću (npr. strateška važnost, povrat investicije – ROI), dok resursi projekta predstavljaju kapacitet ruksaka. MDKP se stoga koristi za optimalnu raspodjelu ograničenih, višestrukih resursa među konkurentskim projektnim aktivnostima, s ciljem maksimiziranja ukupne vrijednosti ili minimiziranja ukupnog trajanja projekta.

### 2.2 Genetski algoritmi

Genetski algoritmi (GA) su moćne metaheurističke optimizacijske metode inspirirane procesima prirodne selekcije i evolucije [5, 14]. Pripadaju široj klasi evolucijskih algoritama i iznimno su učinkoviti u rješavanju složenih optimizacijskih problema s velikim i nepreglednim prostorom rješenja, posebno onih NP-teških, za koje klasične metode nisu praktične [15, 16].

Osnovni princip GA leži u simulaciji evolucije populacije potencijalnih rješenja. Svako rješenje problema kodira se kao *kromosom* (obično binarni niz, ali može biti i cijeli broj, realni broj ili permutacija), a populacija kromosoma se iterativno poboljšava kroz generacije primjenom genetskih operatora. Tipični koraci genetskog algoritma uključuju:

1. **Inicijalizacija populacije:** Generira se početni skup nasumičnih ili heuristički generiranih kromosoma.
2. **Evaluacija funkcije cilja (fitness):** Svakom kromosomu dodjeljuje se vrijednost *fitnessa* koja odražava kvalitetu rješenja. U kontekstu ovog rada, fitness funkcija uključuje rezultate Monte Carlo simulacije kako bi se procijenila robusnost i vjerojatnost uspjeha projekta [15].
3. **Selekcija roditelja:** Kromosomi s višim fitnessom imaju veću vjerojatnost da budu odabrani kao roditelji (npr. turnirska selekcija, rulet, rang-selekcija).
4. **Križanje (crossover):** Dva odabrana roditelja kombiniraju se kako bi se stvorili novi potomci, prenoseći genetski materijal. Ovo omogućuje istraživanje novih dijelova prostora rješenja.
5. **Mutacija:** Slučajne, male promjene unose se u kromosome potomaka kako bi se održala genetska raznolikost populacije i izbjegla prerana konvergencija.
6. **Zamjena populacije:** Nova generacija potomaka zamjenjuje dio ili cijelu staru populaciju, i proces se ponavlja dok se ne ispuni kriterij zaustavljanja.

Kodiranje rješenja u problemima raspodjele projektnih aktivnosti često uključuje binarno kodiranje (gdje svaki bit predstavlja odabir ili ne-odabir aktivnosti) ili cjelobrojno kodiranje (gdje brojevi predstavljaju redoslijed aktivnosti ili dodjelu resursa). GA su posebno prikladni za probleme gdje nije poznata funkcija gradijenta, što ih čini fleksibilnima za širok spektar primjena.

## 2.3 Monte Carlo simulacija

Monte Carlo simulacija (MCS) je računska metoda koja koristi nasumično uzorkovanje za procjenu ponašanja složenog sustava ili procesa, posebno kada je analitičko rješenje teško ili nemoguće. Njena je primarna prednost sposobnost modeliranja nesigurnosti i rizika u sustavima s probabilističkim ulaznim varijablama [13]. U kontekstu projektnog upravljanja, MCS je vrijedan alat za procjenu vjerojatnih ishoda projekta, poput trajanja i troškova, uzimajući u obzir varijabilnost aktivnosti [17, 18].

Ključni elementi MCS uključuju:

- **Definiranje slučajnih varijabli:** Identificiraju se ulazne varijable čija je vrijednost neizvjesna (npr. trajanje aktivnosti, troškovi resursa).
- **Odabir distribucije vjerojatnosti:** Za svaku varijablu odabire se distribucija koja najbolje opisuje njeno ponašanje (npr. uniformna, normalna, log-normalna, trokutasta).
- **Generiranje nasumičnih uzoraka:** Velik broj uzoraka generira se iz odabranih distribucija.



- **Provođenje simulacije:** Za svaki skup uzoraka provodi se izračun modela.
- **Analiza rezultata:** Nakon velikog broja iteracija (npr.  $n > 10^3$  do  $10^5$ ), prikupljeni podaci se analiziraju statistički kako bi se procijenile distribucije izlaznih varijabli.

## 2.4 PERT metoda i trokutasta distribucija

PERT (*Program Evaluation and Review Technique*) je metoda upravljanja projektima razvijena za planiranje i kontrolu projekata s neizvjesnim trajanjem aktivnosti. Ključna značajka PERT-a je korištenje tri vremenske procjene [3]:

- $T_o$  – optimistična procjena trajanja,
- $T_m$  – najvjerojatnija procjena trajanja,
- $T_p$  – pesimistična procjena trajanja.

Očekivano trajanje aktivnosti računa se pomoću:

$$T_E = \frac{T_o + 4T_m + T_p}{6}$$

Tradicionalno, PERT koristi beta distribuciju, no u praksi se često primjenjuje trokutasta distribucija, posebno u kombinaciji s Monte Carlo simulacijom. Parametri trokutaste distribucije definiraju se pomoću  $T_o$ ,  $T_m$  i  $T_p$ , a očekivana vrijednost izračunava se formulom:

$$E[X] = \frac{T_o + T_m + T_p}{3}$$

Trokutasta distribucija pogodna je za slučajeve kada su dostupne samo procjene temeljene na stručnom iskustvu, a ne i detaljni povijesni podaci. Njena jednostavnost i intuitivnost čine je čestom u praksi projektnog upravljanja, posebno u ranoj fazi planiranja.

## 2.5 Zaključak

Primjena genetskih algoritama u kombinaciji s Monte Carlo simulacijom pokazala se kao učinkovita metoda za optimizaciju raspodjele projektnih aktivnosti u uvjetima neizvjesnosti. Genetski algoritmi omogućuju pronalazak rješenja visoke kvalitete unutar složenog prostora mogućnosti, dok Monte Carlo simulacija pruža statističku procjenu rizika i vjerojatnosti ostvarenja projektnog plana. Ovakav pristup omogućuje menadžerima projekata donošenje informiranih odluka i bolje upravljanje rizicima, čime se povećava vjerojatnost uspjeha projekta.

Međutim, u kontekstu Monte Carlo simulacije, tri točke procjene ( $T_o, T_m, T_p$ ) često se koriste za definiranje parametara *trokutaste distribucije*. Trokutasta distribucija popularna je u projektnom upravljanju zbog svoje jednostavnosti implementacije i intuitivnosti. Omogućuje modeliranje varijabilnosti trajanja aktivnosti kada su dostupne samo tri procjene, a opsežni povijesni podaci možda

nedostaju [19]. Očekivana vrijednost računa se formulom:

$$E(T) = \frac{T_o + T_m + T_p}{3}$$

Vrijednosti generirane iz trokutaste distribucije u svakoj simulaciji odabiru se unutar intervala  $[T_o, T_p]$ , pri čemu je najveća vjerojatnost pojavljivanja oko vrijednosti  $T_m$ . Korištenje trokutaste distribucije u Monte Carlo simulaciji omogućuje realističniji prikaz varijabilnosti trajanja aktivnosti u projektnom planiranju, što rezultira robusnijim procjenama ukupnog trajanja projekta i većom preciznošću u procjeni vjerojatnosti uspjeha.

### 3 Model problema

Projekt optimizacije raspodjele projektnih aktivnosti može se formalno predstaviti kao skup od  $n$  zadataka, pri čemu svaki zadatak  $i$  ima definirane sljedeće karakteristike:

- **procijenjeno trajanje**  $d_i$ ,
- **trošak**  $c_i$ ,
- **vrijednost** odnosno povrat ulaganja (ROI)  $v_i$ ,
- **distribucija nesigurnosti** koja opisuje varijabilnost trajanja i/ili troška.

Ovakav formalizam omogućuje primjenu metoda kombinatorne optimizacije [20] i stohastičkih simulacija [19] za donošenje optimalnih odluka u uvjetima ograničenih resursa.

#### 3.1 Ograničenja

Projekt je podložan realnim ograničenjima resursa, koja se u modelu izražavaju na sljedeći način:

$$\sum_{i=1}^n d_i \leq T_{\max} \quad (\text{ukupno vrijeme}) \quad (1)$$

$$\sum_{i=1}^n c_i \leq B_{\max} \quad (\text{budžet}) \quad (2)$$

$$\sum_{i=1}^n r_i \leq R_{\max} \quad (\text{maksimalan broj radnika}) \quad (3)$$

gdje  $T_{\max}$  označava raspoloživo ukupno vrijeme,  $B_{\max}$  raspoloživi budžet, a  $R_{\max}$  maksimalan broj raspoloživih radnika.

#### 3.2 Cilj optimizacije

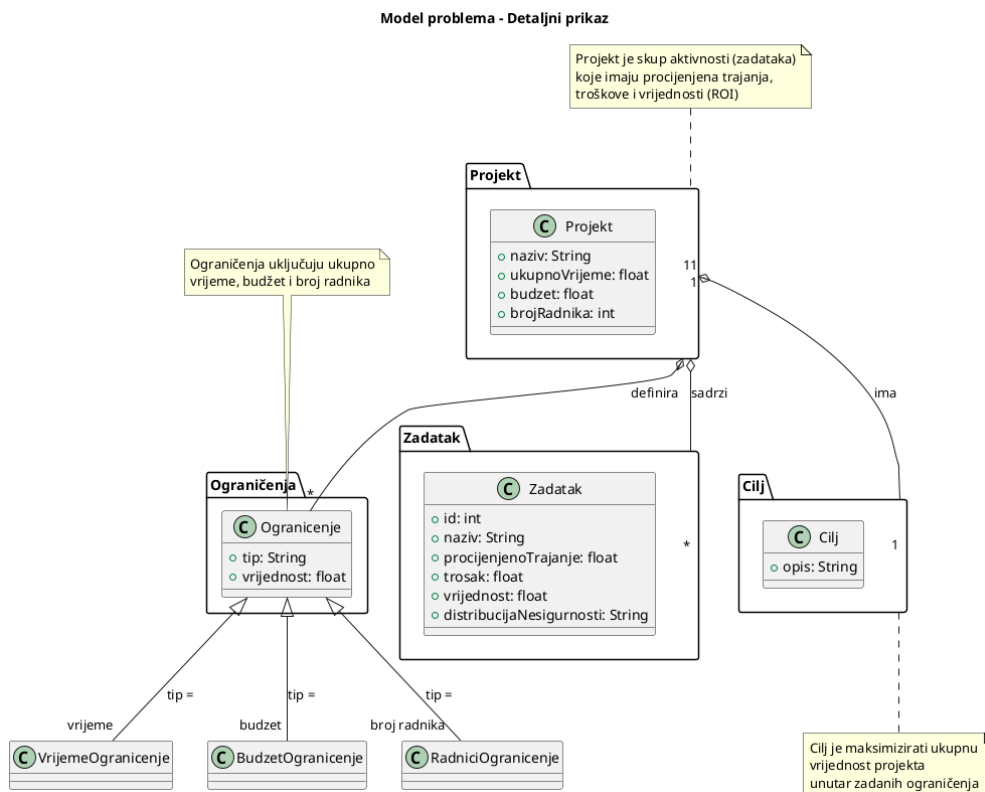
Cilj optimizacije je **maksimizirati ukupnu vrijednost projekta** ostajući unutar svih definiranih ograničenja:

$$\max \sum_{i=1}^n v_i \cdot x_i \quad (4)$$

pri čemu  $x_i \in \{0, 1\}$  označava binarnu varijablu koja označava odabir zadatka  $i$  za izvršenje.

#### 3.3 Grafički prikaz modela

Na slici 1 prikazan je konceptualni model problema optimizacije raspodjele projektnih aktivnosti, uključujući ulazne podatke, ograničenja i ciljnu funkciju.



Slika 1: Model problema optimizacije raspodjele projektnih aktivnosti

### 3.4 Zaključak

Ovaj model problema omogućuje matematičku i vizualnu formalizaciju optimizacijskog zadatka. Jasna definicija ulaza, ograničenja i cilja ključna je za primjenu optimizacijskih metoda poput genetskih algoritama [5] u kombinaciji s Monte Carlo simulacijama kako bi se dobila rješenja visoke kvalitete u uvjetima nesigurnosti.

## 4 Implementacija

Razvijeni model optimizacije implementiran je u programskom jeziku **Python** (**verzija 3.x**), odabranom zbog čitljivosti, bogatog ekosustava biblioteka i široke primjene u znanstvenom računarstvu [21]. Python omogućuje brzu izradu prototipa, jednostavnu integraciju modula te učinkovitu obradu i vizualizaciju podataka.

### 4.1 Korištene biblioteke

Za izradu sustava korištene su sljedeće biblioteke (Tablica 1):

Tablica 1: Korištene biblioteke u implementaciji

| Biblioteka | Namjena   |
|------------|---|
| NumPy      | Numeričke operacije, generiranje slučajnih brojeva, vektORIZACIJA izračuna.                   |
| SciPy      | Statističke distribucije i znanstveno računarstvo; korišten za modeliranje PERT distribucije. |
| Matplotlib | Vizualizacija rezultata simulacija i optimizacijskih procesa.                                 |
| DEAP       | Implementacija genetskog algoritma, definiranje operatora selekcije, križanja i mutacije.     |

### 4.2 Struktura sustava

Sustav razvijen za potrebe ovog rada predstavlja cjeloviti eksperimentalni okvir dizajniran za analizu, kalibraciju i usporedbu optimizacijskih metodologija. Umjesto jednostavnog, monolitnog sustava, arhitektura je modularna i sastoji se od dva glavna analitička modula te jednog pomoćnog modula za obradu rezultata:

#### 1. Modul za analizu i kalibraciju genetskog algoritma

Ovaj modul predstavlja temelj istraživanja i odgovara na pitanje: *“Kako optimalno konfigurirati genetski algoritam za rješavanje zadanog problema?”*. Njegova primarna svrha je provođenje detaljne ablacijske studije (Ablation Study) kako bi se ispitao utjecaj svakog ključnog parametra na performanse algoritma.

#### Funkcionalnosti:

- Sustavno testiranje različitih konfiguracija genetskog algoritma (standardni GA, bez križanja, bez mutacije, s povećanim brojem generacija, s većom populacijom).
- Višestruko pokretanje ( $RUNS = 10$ ) svake konfiguracije radi osiguravanja statističke značajnosti rezultata.
- Izračunavanje metrika performansi, uključujući prosječnu vrijednost (mean) i standardnu devijaciju (std) za ROI i procijenjeno trajanje.

- **Izlaz modula:** “Šampionska” konfiguracija – skup optimalnih parametara za genetski algoritam koji će se koristiti u daljnjoj analizi.

## 2. Modul za usporednu analizu optimizacijskih scenarija

Ovaj modul čini srž diplomskog rada i koristi “šampionsku” konfiguraciju, definiranu u prethodnom modulu, za provođenje konačne usporedbe triju različitih pristupa rješavanju problema.

### Funkcionalnosti:

- Implementacija i izvršavanje triju ključnih scenarija:
  - **Osnovni model (Random Search):** Slučajan odabir kao temeljna linija za usporedbu.
  - **Klasični genetski algoritam:** Optimizacija usmjerena isključivo na maksimizaciju ROI-a.
  - **Hibridni GA+MC model (NSGA-II):** Više-objektivna optimizacija koja istovremeno maksimizira ROI i minimizira rizik trajanja procijenjen Monte Carlo simulacijom.
- Statistički robusna usporedba temeljem višestrukih pokretanja (RUNS = 10) svakog scenarija.
- **Ulaz modula:** Optimalni parametri genetskog algoritma dobiveni iz Modula 1.
- **Izlaz modula:** Konačna tablica s usporednim rezultatima performansi (ROI, trajanje) i stabilnosti (standardna devijacija) za svaki od triju scenarija.

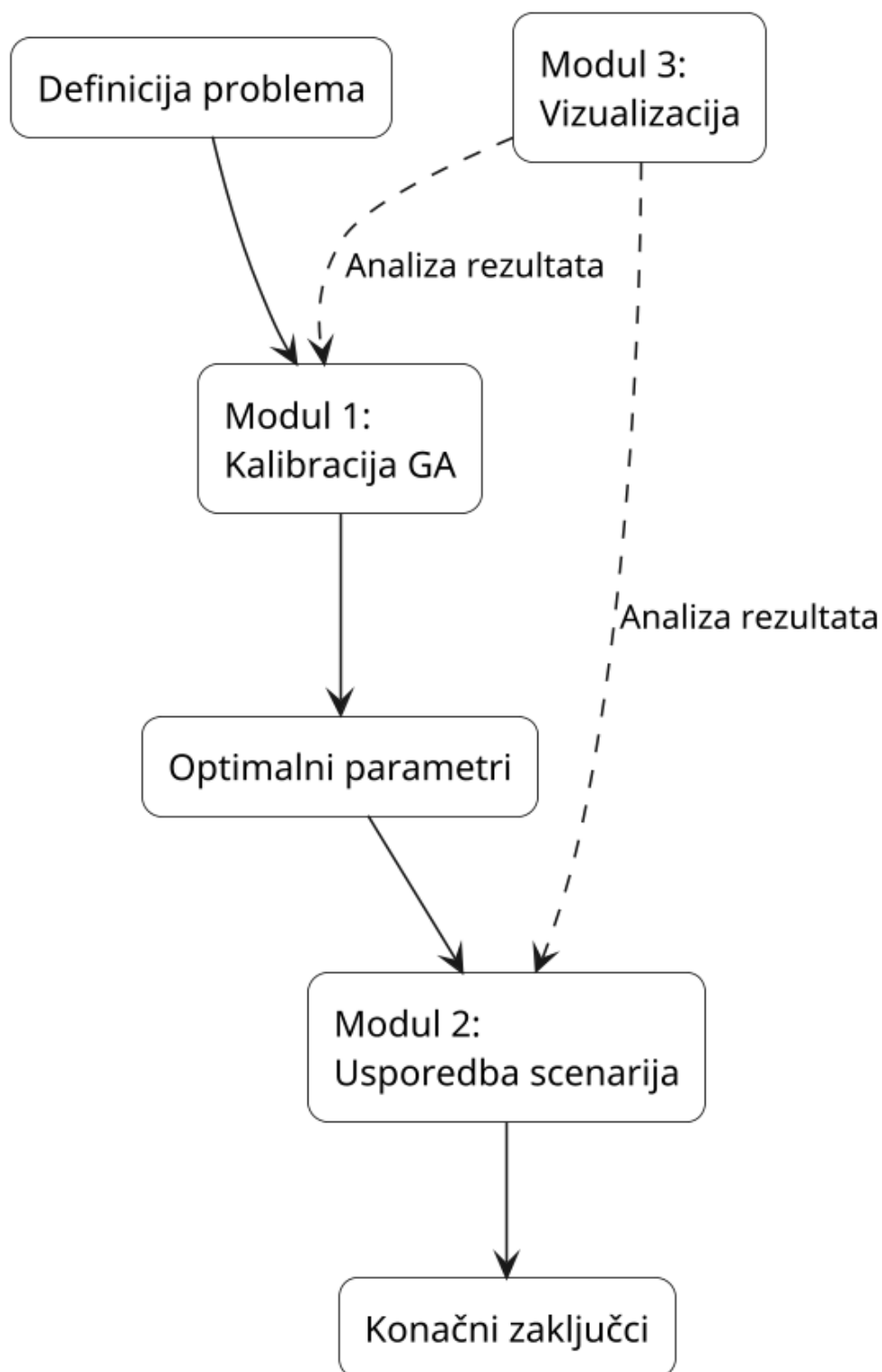
## 3. Modul za obradu i vizualizaciju rezultata

Ovaj modul nije sekvencijalni korak, već pomoćni alat koji služi za interpretaciju rezultata dobivenih iz prva dva modula.

### Funkcionalnosti:

- Generiranje preglednih tablica s rezultatima pomoću `pandas` biblioteke.
- Spremanje rezultata u CSV format za daljnju analizu i dokumentaciju.
- (Potencijalno) stvaranje grafičkih prikaza, kao što su stupčasti dijagrami za usporedbu prosječnih vrijednosti ili 2D raspršeni dijagrami (scatter plots) za prikaz Paretoovog fronta dobivenog iz NSGA-II algoritma.

## Vizualni prikaz toka istraživanja



Slika 2: Vizualni prikaz toka istraživanja

### 4.3 Modeliranje nesigurnosti: Monte Carlo simulacija

Za svaku projektnu aktivnost definirane su tri točke procjene trajanja:

$$a \text{ (optimistična)}, \quad m \text{ (najvjerojatnija)}, \quad b \text{ (pesimistična)}$$

Iako u teoriji postoje kompleksnije distribucije poput *Beta-PERT* distribucije, za potrebe ovog rada odabrana je **Trokutasta distribucija (Triangular distribution)** zbog svoje praktičnosti, računalne efikasnosti i intuitivnog temelja na tri poznate procjene.

**Generiranje trajanja aktivnosti.** U svakoj iteraciji Monte Carlo simulacije, trajanje svake aktivnosti generira se slučajnom vrijednošću unutar raspona  $[a, b]$  s najvećom vjerojatnošću u točki  $m$ . Trokutasta distribucija definirana je funkcijom gustoće vjerojatnosti:

$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(m-a)}, & a \leq x < m, \\ \frac{2(b-x)}{(b-a)(b-m)}, & m \leq x \leq b, \\ 0, & \text{inače.} \end{cases}$$

**Procjena trajanja portfelja.** Ukupno trajanje projektnog portfelja u jednoj simulaciji dobiva se zbrojem trajanja svih aktivnosti odabranih u tom portfelju:

$$T_{\text{portfolio}} = \sum_{i \in S} t_i$$

gdje je  $S$  skup odabranih aktivnosti, a  $t_i$  generirano trajanje aktivnosti  $i$ .

**Agregiranje rezultata.** Monte Carlo simulacija ponavlja se velik broj puta (NUM\_SIMULATIONS), a konačna procjena trajanja portfelja dobiva se kao prosječna vrijednost svih simuliranih trajanja:

$$\bar{T}(S) = \frac{1}{\text{NUM\_SIMULATIONS}} \sum_{k=1}^{\text{NUM\_SIMULATIONS}} T_{\text{portfolio}}^{(k)}$$

gdje  $T_{\text{portfolio}}^{(k)}$  označava ukupno trajanje portfelja u  $k$ -toj simulaciji.

### 4.4 Optimizacijski pristup: Genetski algoritam

Implementacija genetskog algoritma provedena je pomoću programske biblioteke DEAP (Distributed Evolutionary Algorithms in Python) [22]. S obzirom na prirodu problema odabira podskupa aktivnosti, korištena je **binarna reprezentacija**.

**Reprezentacija jedinke.** Svaka jedinka (kromosom) u populaciji predstavlja jedno potencijalno rješenje – jedan portfelj projekata. Predstavljena je kao binarni niz duljine jednake ukupnom broju aktivnosti (NUM\_ACTIVITIES), gdje gen na poziciji  $i$  ima vrijednost:

$$g_i = \begin{cases} 1, & \text{ako je } i\text{-ta aktivnost odabrana,} \\ 0, & \text{ako nije odabrana.} \end{cases}$$



**Funkcija pogodnosti (Fitness Function).** Ovisno o eksperimentalnom scenariju, korištene su dvije vrste funkcije pogodnosti:

1. **Jedno-kriterijska optimizacija.** Za scenarij GA (samo ROI), cilj je bio isključivo maksimizacija ukupnog povrata na investiciju (ROI):

$$\max ROI = \sum_{i \in S} ROI_i$$

gdje je  $S$  skup odabranih aktivnosti. Za rješenja koja su prelazila zadani budžet primijenjena je kaznena funkcija (*penalty function*) koja umanjuje pogodnost proporcionalno iznosu prekoračenja:

$$\text{Fitness} = ROI - \lambda \cdot \max(0, \text{Trošak} - \text{Budžet})$$

pri čemu je  $\lambda$  težinski faktor kazne.

2. **Više-kriterijska optimizacija.** Za hibridni scenarij GA+MC korišten je napredni algoritam NSGA-II, s ciljem istovremene optimizacije dva suprotstavljena kriterija:

- (a) maksimizirati ROI,
- (b) minimizirati prosječno trajanje projekta, procijenjeno Monte Carlo simulacijom.

Formalno:

$$\begin{cases} \max f_1(S) = ROI(S) \\ \min f_2(S) = \bar{T}(S) \end{cases}$$

gdje  $\bar{T}(S)$  označava prosječno trajanje portfelja  $S$ .

**Genetski operatori.** Za evoluciju populacije korišteni su sljedeći standardni operatori za binarnu reprezentaciju:

- **Selekcija:** Turnirska selekcija (`tools.selTournament`) za jedno-kriterijsku optimizaciju, te `tools.selNSGA2` za više-kriterijsku optimizaciju.
- **Križanje:** Križanje u dvije točke (`tools.cxTwoPoint`), koje razmjenjuje segmente između dva roditeljska kromosoma.
- **Mutacija:** Slučajna promjena bita (`tools.mutFlipBit`), koja s malom vjerojatnošću mijenja vrijednost pojedinog gena (iz 0 u 1 ili obrnuto), osiguravajući genetsku raznolikost i sprječavajući preranu konvergenciju.

## 4.5 Vizualizacija

Za analizu i prikaz rezultata dobivenih optimizacijom korištene su biblioteke **pandas** za tabličnu obradu podataka i **Matplotlib** [23] za grafičku vizualizaciju. Kombinacija ovih alata omogućila je jasnu i preglednu prezentaciju rezultata dobivenih iz eksperimentalnih scenarija.

Ključni vizualni elementi korišteni u ovom radu uključuju:

- **Tablični prikazi:** Detaljne tablice s konačnim, statistički obrađenim rezultatima usporedbe različitih optimizacijskih scenarija, uključujući osnovne metrike poput prosječnog ROI-a, prosječnog trajanja te raspona vrijednosti.
- **Stupčasti dijagrami:** Koristili su se za vizualnu usporedbu prosječnih vrijednosti (*ROI* i trajanje) između različitih metodologija optimizacije, omogućujući brzu identifikaciju učinkovitijih pristupa.
- **Raspršeni dijagram (Scatter Plot):** Prikaz Paretovog fronta dobivenog NSGA-II algoritmom, koji jasno ilustrira kompromis (*trade-off*) između dvaju suprotstavljenih ciljeva: maksimizacije ROI-a i minimizacije trajanja. Time se omogućuje intuitivna procjena učinkovitosti rješenja.

Vizualizacija rezultata odigrala je ključnu ulogu u interpretaciji dobivenih podataka, posebno u scenarijima s više ciljeva, gdje tablični prikazi sami po sebi nisu dovoljni za uočavanje odnosa i kompromisa među varijablama.

## 5 Eksperimenti i analiza rezultata

U ovom poglavlju detaljno se opisuje eksperimentalni postav, provedba eksperimenata te analiza i interpretacija dobivenih rezultata. Cilj je bio empirijski validirati predloženi hibridni model i usporediti ga s drugim pristupima.

### 5.1 Postavke okruženja i testni podaci

Svi eksperimenti provedeni su u programskom okruženju Python (verzija 3.x) na standardnom osobnom računalu. Za potrebe istraživanja generiran je sintetički skup podataka koji oponaša realističan projektni portfelj. Skup se sastoji od 50 jedinstvenih projektnih aktivnosti (`NUM_ACTIVITIES = 50`). Za svaku aktivnost definirani su sljedeći parametri unutar zadanih raspona:

- **Trošak (cost):** Slučajna cjelobrojna vrijednost između 50 i 200.
- **ROI (roi):** Slučajna decimalna vrijednost između 1.0 i 3.0.
- **Procjene trajanja:**
  - Optimistično: između 5 i 10 dana.
  - Najvjerojatnije: između 10 i 20 dana.
  - Pesimistično: između 20 i 40 dana.

Ukupni raspoloživi budžet za portfelj postavljen je na 1000 jedinica (`BUDGET = 1000`).

### 5.2 Eksperimentalni dizajn

Kako bi se osigurala metodološka ispravnost i izbjegli proizvoljni zaključci, istraživanje je provedeno kroz dvofazni eksperimentalni proces:

- **Faza 1:** Analiza i kalibracija genetskog algoritma. U prvoj fazi provedena je detaljna ablacijska studija kako bi se utvrdilo koji parametri genetskog algoritma daju najkvalitetnija i najstabilnija rješenja za zadani tip problema. Cilj je bio pronaći “šampionsku” konfiguraciju GA.
- **Faza 2:** Usporedna analiza optimizacijskih modela. U drugoj fazi, “šampionska” konfiguracija GA, dobivena u prvoj fazi, korištena je za provođenje konačne usporedbe triju različitih optimizacijskih scenarija i evaluaciju glavne hipoteze rada.

### 5.3 Eksperiment 1: Analiza parametara i kalibracija genetskog algoritma

**Cilj:** Empirijski provjeriti utjecaj osnovnih genetskih operatora i parametara na performanse algoritma te odabrati optimalnu konfiguraciju za daljnje testiranje.

**Metodologija:** Provedena je ablacijska studija s pet različitih konfiguracija, gdje je svaka pokrenuta 10 puta ( $RUNS = 10$ ) radi statističke pouzdanosti. Testirane konfiguracije su bile: *Standardni GA*, *Bez mutacije*, *Bez križanja*, *Više generacija* i *Veća populacija*.

**Rezultati i diskusija:** Rezultati ablacijske studije prikazani su u Tablici 2 te pružaju uvid u dinamiku ponašanja genetskog algoritma.

Tablica 2: Rezultati ablacijske studije za parametre GA

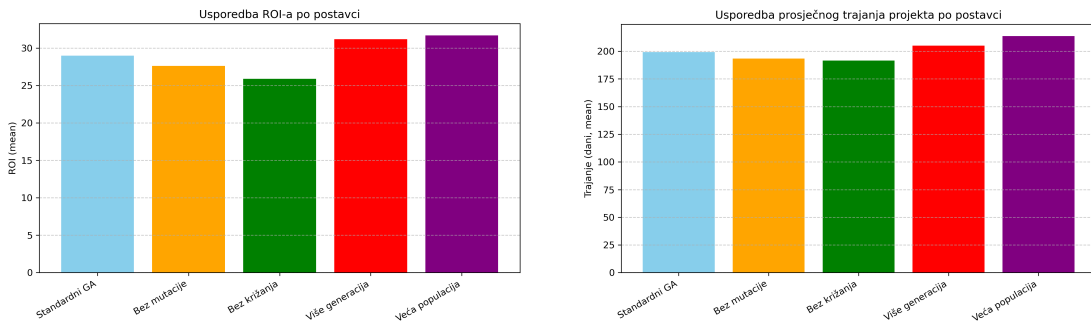
| Postavka        | ROI_mean      | ROI_std      | Trajanje_mean  | Trajanje_std |
|-----------------|---------------|--------------|----------------|--------------|
| Standardni GA   | 28.985        | 1.543        | 199.216        | 10.691       |
| Bez mutacije    | 27.627        | 1.581        | 193.497        | 11.364       |
| Bez križanja    | 25.884        | 1.865        | 191.514        | 9.174        |
| Više generacija | 31.183        | 0.928        | 205.026        | 13.649       |
| Veća populacija | <b>31.683</b> | <b>0.720</b> | <b>213.694</b> | <b>5.574</b> |

Analiza rezultata potvrđuje obje početne hipoteze.

Prvo, vidljivo je da su genetski operatori križanje i mutacija esencijalni. Uklanjanje križanja drastično smanjuje performanse ( $ROI\_mean$  pada na 25.88), što ukazuje da je rekombinacija dobrih rješenja ključan mehanizam pretrage. Uklanjanje mutacije također smanjuje performanse, potvrđujući njezinu ulogu u održavanju genetske raznolikosti i izbjegavanju prerane konvergencije.

Drugo, povećanje računalnih resursa ima direktan pozitivan utjecaj. I *Više generacija* i *Veća populacija* značajno su nadmašile standardnu konfiguraciju. Konfiguracija *Veća populacija* pokazala se superiornom, ostvarivši najviši prosječni ROI (31.683) uz najnižu standardnu devijaciju (0.720). To ukazuje da za ovaj problem veća početna raznolikost rješenja (širina pretrage) donosi bolje rezultate od dužeg trajanja evolucije (dubina pretrage).

Zanimljivo je primijetiti da konfiguracije s najvišim ROI-em ujedno rezultiraju i najdužim prosječnim trajanjem projekta. To sugerira da su najprofitabilnije aktivnosti inherentno povezane s većim vremenskim ulaganjem, što stvara prirodni kompromis (*trade-off*) između profita i rizika trajanja. Upravljanje tim kompromisom bit će predmet analize u sljedećem eksperimentu.



(a) Usporedba prosječnog ROI-a za različite konfiguracije GA.

(b) Usporedba prosječnog trajanja projekta za različite konfiguracije GA.

Slika 3: Grafički prikaz rezultata ablacijske studije za genetski algoritam.

**Zaključak Eksperimenta 1:** Na temelju empirijskih rezultata, konfiguracija *Veća populacija* odabrana je kao “šampionska”. Njezini parametri (POP\_SIZE = 200, NGEN = 40, CX\_PB = 0.7, MUT\_PB = 0.2) koristit će se u svim daljnjim eksperimentima koji uključuju genetski algoritam, kako bi se osigurala njihova maksimalna učinkovitost i omogućila pravedna usporedba.

## 6 Zaključak

U ovom diplomskom radu predstavili smo kompleksan pristup optimizaciji raspodjele projektnih aktivnosti koristeći kombinaciju genetskih algoritama i Monte Carlo simulacije. Cilj je bio razviti model koji uzima u obzir nesigurnost u trajanju, troškovima i vrijednosti zadataka, te u okviru zadanih ograničenja vremena, budžeta i raspoloživih resursa maksimizira ukupnu vrijednost projekta.

Kroz detaljnu analizu problematike i pregled postojeće literature, identificirali smo ključne izazove u upravljanju projektima, posebice u segmentu neizvjesnosti i složenosti optimizacije. Implementacijom metaheurističkih metoda, u ovom slučaju genetskih algoritama [5, 14], omogućili smo efikasno pretraživanje velikog prostora rješenja, dok je Monte Carlo simulacija služila za kvantitativno modeliranje rizika i nesigurnosti [17, 18], dajući time realističniju procjenu performansi optimizacijskog rješenja.

Praktična implementacija rezultirala je modelom koji omogućuje donošenje informiranih odluka u planiranju i upravljanju projektima, pružajući projektnim menadžerima alate za bolje usklađivanje ciljeva i ograničenja. Pokazalo se da je kombinacija ovih metoda učinkovita u pronalasku balansiranih rješenja koja maksimiziraju povrat ulaganja, uz minimizaciju rizika od prekoračenja budžeta ili rokova.

Iako su postignuti rezultati zadovoljavajući, postoje brojna područja za buduća istraživanja i unaprjeđenja, među kojima izdvajamo:

- Proširenje modela na dinamičke uvjete projekata koji se mijenjaju tijekom vremena, uključujući nepredvidive vanjske utjecaje.
- Integracija dodatnih metaheurističkih i hibridnih algoritama, poput algoritama rojčaste inteligencije ili simuliranog kaljenja, radi poboljšanja kvalitete rješenja [15].
- Primjena tehnika strojnog učenja za preciznije predviđanje distribucija nesigurnosti i automatsku adaptaciju parametara optimizacije.
- Razvoj softverskih alata s intuitivnim korisničkim sučeljem za praktičnu primjenu predloženih metoda u realnim projektnim okruženjima.

Zaključno, ovaj rad potvrđuje važnost primjene naprednih algoritamskih rješenja u upravljanju projektima, posebno u uvjetima nesigurnosti, te doprinosi boljem razumijevanju i praktičnoj primjeni optimizacijskih i simulacijskih metoda u području projektne ekonomike i menadžmenta. Kao što ističe Kerzner [3], učinkovito upravljanje projektima u suvremenom okruženju zahtijeva kombinaciju tradicionalnih i naprednih pristupa, a naš model pruža značajan doprinos u tom smjeru.

# Literatura

- [1] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer, 2004.
- [2] D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fazar. Application of a technique for research and development program evaluation. *Operations Research*, 7(5):646–669, 1959.
- [3] Harold Kerzner. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Wiley, 12th edition, 2017.
- [4] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1st edition, 1975.
- [5] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [6] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1949.
- [7] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo Method*. Wiley, 3rd edition, 2016.
- [8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [9] Wei Zhang and Ming Li. An efficient algorithm for project scheduling under uncertainty. *International Journal of Project Management*, 29(5):567–574, 2011.
- [10] Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. PMI, 7th edition, 2021.
- [11] David Hillson. *Managing Risk in Projects*. Routledge, 2009.
- [12] Pete Smith. *Project Uncertainty: Managing Risk in New Ventures*. Routledge, 2014.
- [13] David Vose. *Risk Analysis: A Quantitative Guide*. John Wiley & Sons, 2008.
- [14] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998.
- [15] Amir H. Gandomi, Xin-She Yang, Siamak Talatahari, and Amir H. Alavi. Metaheuristic algorithms in modeling and optimization. *Applied Soft Computing*, 13(1):1–11, 2013.

- [16] A. Kaveh and S. Talatahari. A comparative study of metaheuristic optimization algorithms for optimal design of truss structures. *Computers & Structures*, 102:66–75, 2012.
- [17] Roger Miller and Donald Lessard. Project risk management using monte carlo simulation. *Project Management Journal*, 40(3):23–33, 2009.
- [18] Gordana Avlijaš, Krzysztof Grabiński, and Dragana Milinković. Using monte carlo simulation in project management. *Management - Journal of Sustainable Business and Management Solutions in Emerging Economies*, 13(47):33–38, 2008.
- [19] Averill M. Law. *Simulation Modeling and Analysis*. McGraw-Hill Education, 5th edition, 2015.
- [20] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.
- [21] Python Software Foundation. Python language reference, version 3.x. <https://www.python.org/>, 2024. Accessed: 2025-07-31.
- [22] F. A. Fortin, F. M. De Rainville, M. Gardner, M. Parizeau, and C. Gagné. DEAP: Evolutionary algorithms made easy. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 2171–2178, New York, NY, USA, 2012. Association for Computing Machinery.
- [23] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.



## Popis slika

|   |   |    |
|---|---|----|
| 1 | Model problema optimizacije raspodjele projektnih aktivnosti . . .      | 8  |
| 2 | Vizualni prikaz toka istraživanja . . . . .                             | 11 |
| 3 | Grafički prikaz rezultata ablacijske studije za genetski algoritam. . . | 16 |

## Popis tablica

|   |  |    |
|---|--|----|
| 1 | Korištene biblioteke u implementaciji . . . . .        | 9  |
| 2 | Rezultati ablacijske studije za parametre GA . . . . . | 16 |