

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

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 [?]. 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*) [?, ?] 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 [?, ?], dok Monte Carlo simulacije (MC) omogućuju modeliranje i kvantifikaciju neizvjesnosti [?, ?]. Međutim, u literaturi postoji ograničen broj radova koji integriraju ove dvije metode u jedinstven hibridni pristup optimizaciji projektnih aktivnosti [?, ?]. Praznina se očituje u nedostatku sustavnog eksperimentalnog okvira za kalibraciju i usporednu analizu hibridnih GA-MC modela, čime bi se precizno kvantificiralo kako takva sinergija utječe na robusnost rješenja

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 [?]. 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 [?].

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 [?], dok Monte Carlo simulacije omogućuju procjenu varijabilnosti i rizika rješenja [?]. Kombiniranjem ovih metoda moguće je razviti pristup koji generira kvalitetna rješenja, a pritom osigurava njihovu robusnost u uvjetima nesigurnosti [?].

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 [?]. Oni mogu biti tehničke, organizacijske, financijske ili tržišne prirode, a često su međusobno povezani [?]. Nesigurnosti proizlaze iz nepotpunih informacija, promjenjivih uvjeta i nemogućnosti preciznog predviđanja budućih događaja [?]. Upravljanje rizicima uključuje njihovu identifikaciju, procjenu i razvoj strategija odgovora [?].

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

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

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. U kojim uvjetima i prema kojim metrikama (npr. profitabilnost, rizik trajanja, stabilnost) hibridni GA-MC pristup postiže superiorne rezultate u odnosu na samostalnu primjenu GA ili MC modela?
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 [?], dok MC kvantificira rizik i procjenjuje varijabilnost tih rješenja kroz stohastičko modeliranje [?]. 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** detaljno opisuje matematički i konceptualni model problema.
- **Poglavlje 4** prikazuje metodologiju i arhitekturu razvijenog softverskog sustava, uključujući implementaciju Monte Carlo simulacije i genetskog algoritma.

- **Poglavlje 5** opisuje dvo-fazni eksperimentalni dizajn, analizu dobivenih rezultata i diskusiju o performansama testiranih modela.
- **Poglavlje 6** sadrži zaključak rada, sažetak doprinosa, osvrt na ograničenja 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 metode procjene trajanja 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 [?]. 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:

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

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, projektne aktivnosti se mogu interpretirati kao objekti s određenom vrijednošću (npr. strateška važnost, povrat investicije – ROI), dok svaka aktivnost troši više vrsta resursa (npr. vrijeme, budžet, broj radnika), koji predstavljaju različite dimenzije "težine". Kapacitet ruksaka tada predstavlja ukupnu raspoloživost svakog od resursa. MDKP se stoga koristi kao snažan model za optimalnu raspodjelu ograničenih, višestrukih resursa među konkurentskim projektnim aktivnostima.

2.2 Genetski algoritmi

Genetski algoritmi (GA) su moćne metaheurističke optimizacijske metode inspirirane procesima prirodne selekcije i evolucije [?, ?]. 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 [?, ?].

Osnovni princip GA leži u simulaciji evolucije populacije potencijalnih rješenja. Svako rješenje problema kodira se kao kromosom (u ovom radu, binarni niz), a populacija kromosoma se iterativno poboljšava kroz generacije primjenom genetskih operatora. Tijek genetskog algoritma uključuje:

1. **Inicijalizacija populacije:** Generira se početni skup nasumičnih kromosoma.

2. **Evaluacija funkcije cilja (fitness):** Svakom kromosomu dodjeljuje se vrijednost pogodnosti (fitnessa) koja odražava kvalitetu rješenja. U ovisnosti o cilju, funkcija pogodnosti može biti jedno-kriterijska (npr. maksimizacija ROI-a) ili više-kriterijska. U naprednijim modelima, kao što je hibridni model razvijen u ovom radu, fitness funkcija može uključivati i rezultate Monte Carlo simulacije kako bi se procijenila robusnost rješenja.
3. **Selekcija roditelja:** Kromosomi s višim fitnessom imaju veću vjerojatnost da budu odabrani kao roditelji za stvaranje sljedeće generacije.
4. **Križanje (crossover):** Dva odabrana roditelja kombiniraju se kako bi se stvorili novi potomci, prenoseći genetski materijal i istražujući nove dijelove 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 (npr. zadan broj generacija).

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 [?]. U kontekstu projektnog upravljanja, MCS je vrijedan alat za procjenu vjerojatnih ishoda projekta, poput trajanja i troškova, uzimajući u obzir varijabilnost aktivnosti [?, ?].

Ključni elementi MCS uključuju:

- **Definiranje slučajnih varijabli:** Identificiraju se ulazne varijable čija je vrijednost neizvjesna (npr. trajanje aktivnosti).
- **Odabir distribucije vjerojatnosti:** Za svaku varijablu odabire se distribucija koja najbolje opisuje njeno ponašanje.
- **Generiranje nasumičnih uzoraka:** Velik broj uzoraka generira se iz odabranih distribucija.
- **Provođenje simulacije:** Za svaki skup uzoraka provodi se izračun modela (npr. zbrajanje trajanja aktivnosti).
- **Analiza rezultata:** Nakon velikog broja iteracija, prikupljeni podaci se analiziraju statistički kako bi se dobila distribucija mogućih ishoda.

2.4 Modeliranje nesigurnosti trajanja: PERT i Trokutasta distribucija

Metodologija PERT (Program Evaluation and Review Technique) uvela je praksu korištenja tri vremenske procjene za aktivnosti s neizvjesnim trajanjem [?]:

- T_o – optimistična procjena trajanja (najkraće moguće trajanje).
- T_m – najvjerojatnija procjena trajanja (očekivano trajanje).
- T_p – pesimistična procjena trajanja (najduže moguće trajanje).

Dok tradicionalna PERT metoda koristi ove tri točke za izračun parametara Beta distribucije, u modernoj praksi upravljanja rizikom, a posebno u Monte Carlo simulacijama, često se koristi Trokutasta distribucija zbog svoje jednostavnosti i intuitivnosti [?].

Trokutasta distribucija je kontinuirana distribucija vjerojatnosti definirana s tri parametra: minimum (a), maksimum (b) i najvjerojatnija vrijednost (c), što direktno odgovara procjenama T_o , T_p i T_m . Njena je glavna prednost što ne zahtijeva opsežne povijesne podatke, već se može temeljiti na stručnom iskustvu, što je čini iznimno pogodnom za projektno planiranje. Slučajne vrijednosti generirane iz ove distribucije nalaze se unutar intervala $[T_o, T_p]$, s najvećom vjerojatnošću pojavljivanja oko vrijednosti T_m . Prosječna vrijednost (očekivano trajanje) za Trokutastu distribuciju računa se jednostavnom formulom:

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

Upravo je Trokutasta distribucija, zbog navedenih prednosti, odabrana kao temelj za modeliranje nesigurnosti trajanja aktivnosti u Monte Carlo simulacijama provedenim u ovom radu.

3 Model problema

U ovom poglavlju formalno se definira problem optimizacije portfelja projektnih aktivnosti. Precizno se opisuju karakteristike aktivnosti, primijenjena ograničenja te jedno-kriterijski i više-kriterijski ciljevi optimizacije koji su korišteni u eksperimentalnoj evaluaciji.

3.1 Formalna definicija problema

Problem se definira kao odabir optimalnog podskupa (portfelja) aktivnosti iz većeg, unaprijed definiranog skupa dostupnih aktivnosti, što je čest problem u projektnom menadžmentu [?, ?].

Neka je $A = \{a_1, a_2, \dots, a_n\}$ skup od n dostupnih projektnih aktivnosti. Za svaku aktivnost $a_i \in A$ definirani su sljedeći atributi:

- **Trošak (c_i):** Količina budžeta potrebna za izvođenje aktivnosti.
- **Vrijednost (v_i):** Povrat na investiciju (ROI) koji se ostvaruje uspješnim završetkom aktivnosti.
- **Nesigurnost trajanja (T_o, T_m, T_p):** Trajanje aktivnosti nije deterministička vrijednost, već stohastička varijabla opisana s tri točke procjene: optimističnom, najvjerojatnijom i pesimističnom, koje služe kao parametri za Trokutastu distribuciju u Monte Carlo simulaciji.

Cilj je definirati binarni vektor odluke $x = (x_1, x_2, \dots, x_n)$, gdje $x_i \in \{0, 1\}$. Ako je $x_i = 1$, aktivnost a_i je odabrana za uključivanje u portfelj; ako je $x_i = 0$, aktivnost se ne izvodi.

3.2 Ograničenja modela

Iako realni projekti mogu imati višestruka ograničenja, u ovom modelu implementirano je ključno i najčešće ograničenje u upravljanju portfeljem:

- **Ograničenje budžeta (B_{max}):** Ukupni zbroj troškova svih odabranih aktivnosti ne smije prelaziti raspoloživi budžet. Formalno ovo ograničenje odgovara klasičnoj formulaciji problema ruksaka (Knapsack Problem) [?]:

$$\sum_{i=1}^n c_i x_i \leq B_{max}$$

Vrijedi napomenuti da ukupno trajanje portfelja nije tretirano kao strogo ograničenje, već kao izlazna metrika performansi i cilj za minimizaciju. Ovakav pristup je fleksibilniji i realističniji, jer menadžerima često nije cilj samo "uklopiti se" u zadani rok, već pronaći portfelj s najboljim mogućim očekivanim trajanjem za određenu razinu povrata na investiciju, što je u skladu s modernim praksama upravljanja projektnom nesigurnošću [?].

3.3 Ciljevi optimizacije

U skladu s eksperimentalnim dizajnom, definirana su dva različita optimizacijska cilja koja odgovaraju testiranim scenarijima:

- **Jedno-kriterijski cilj: Maksimizacija povrata na investiciju (ROI)**
Ovaj cilj odgovara klasičnom GA (samo ROI) modelu. Ciljna funkcija je maksimizacija ukupnog zbroja ROI vrijednosti odabranih aktivnosti, uz poštivanje ograničenja budžeta. Ovakav tip optimizacijskog cilja čest je u primjeni genetskih algoritama na probleme alokacije resursa [?].

$$\max \sum_{i=1}^n v_i x_i$$

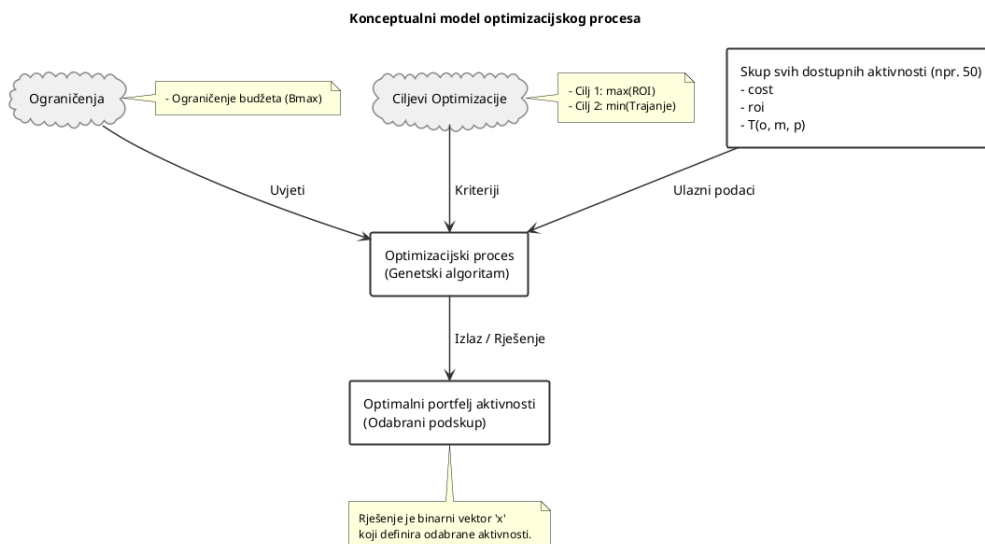
- **Više-kriterijski cilj: Maksimizacija ROI-a i minimizacija trajanja**
Ovaj cilj odgovara naprednom hibridnom GA+MC (NSGA-II) modelu i predstavlja srž istraživanja. Ovdje se istovremeno optimiziraju dva suprotstavljena cilja, što zahtijeva primjenu specijaliziranih više-kriterijskih algoritama. U ovom radu korišten je NSGA-II algoritam, koji je jedan od najpoznatijih i najčešće korištenih u toj domeni [?]:

1. **Cilj 1 (Profitabilnost):** Maksimizirati ukupni ROI ($\max \sum v_i x_i$).
2. **Cilj 2 (Rizik):** Minimizirati očekivano ukupno trajanje portfelja ($\min E[T(x)]$), gdje je $E[T(x)]$ prosječno trajanje dobiveno Monte Carlo simulacijom za odabrani portfelj x .

Ovakav više-kriterijski pristup ne traži jedno jedino "najbolje" rješenje, već skup optimalnih kompromisnih rješenja (Pareto front).

3.4 Konceptualni model

Konceptualni model problema, koji prikazuje proces odabira optimalnog portfelja iz skupa dostupnih aktivnosti pod utjecajem ograničenja i optimizacijskih ciljeva, prikazan je na Slici ??.



Slika 1: Konceptualni model problema optimizacije portfelja projektnih aktivnosti

Ovaj model problema omogućuje jasnu matematičku i vizualnu formalizaciju optimizacijskog zadatka. Precizna definicija ulaza, ograničenja i ciljeva ključna je za primjenu optimizacijskih metoda poput genetskih algoritama [?] u kombinaciji s Monte Carlo simulacijama [?], kako bi se dobila rješenja visoke kvalitete koja su istovremeno i robusna na prisutne 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 [?]. 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 ??):

Tablica 1: Korištene biblioteke u implementaciji

Biblioteka	Namjena i citat
Python	Osnovni programski jezik za cjelokupnu implementaciju. [?]
DEAP	Okvir za razvoj i provedbu evolucijskih algoritama. [?]
NumPy	Numeričke operacije i statistička obrada nizova podataka. [?]
Pandas	Učitavanje, obrada i spremanje tabličnih podataka s rezultatima. [?]
Seaborn	Kreiranje naprednih statističkih vizualizacija (stupčasti, linijski i raspršeni grafikoni). [?]
Matplotlib	Osnovna biblioteka za crtanje na koju se oslanja Seaborn. [?]
Random	Standardna Python biblioteka korištena za generiranje slučajnih brojeva i uzorkovanje iz Trokutaste distribucije.

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 ispitaio 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 osiguranja 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.

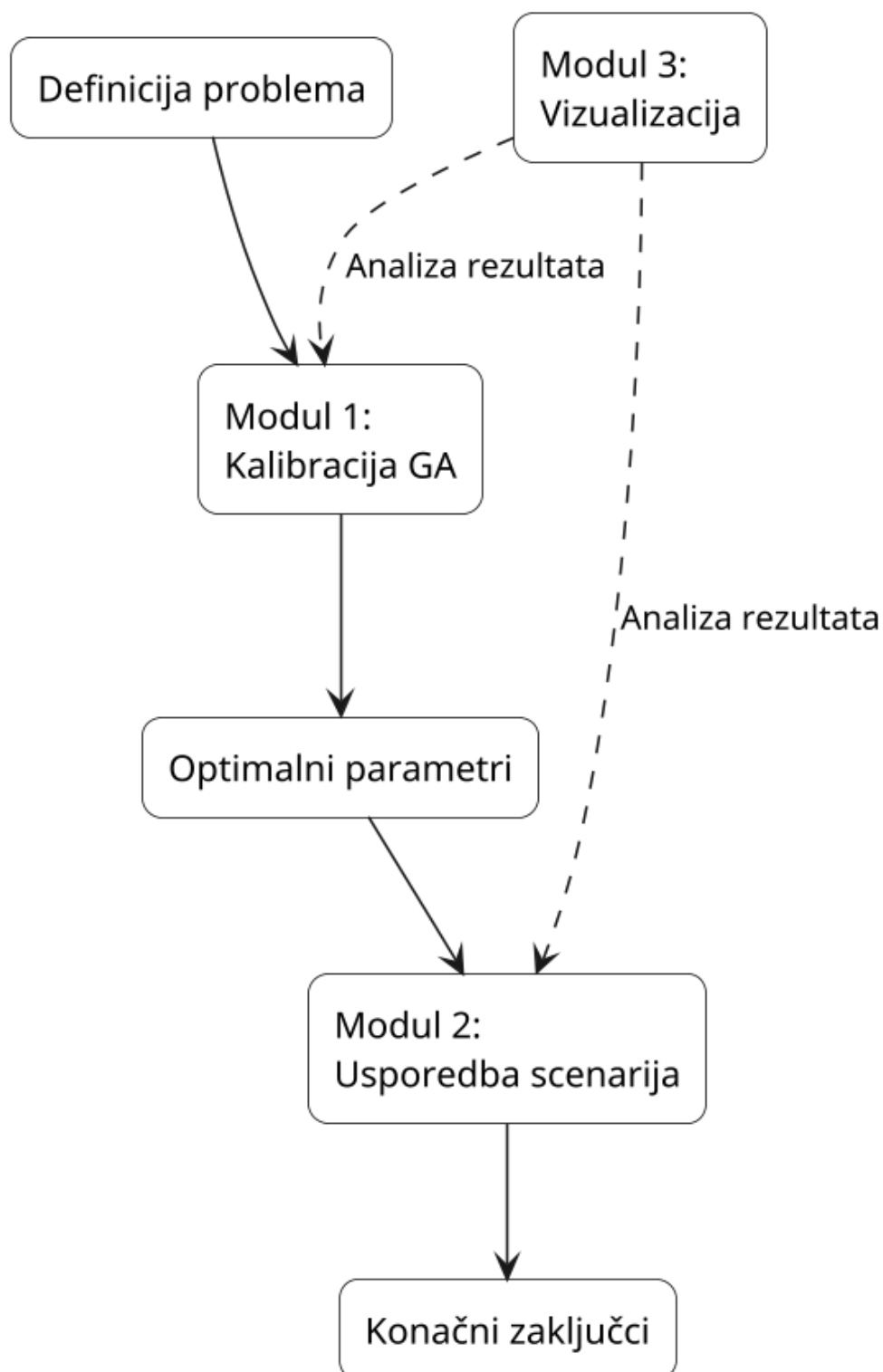
3. Modul za obradu i vizualizaciju rezultata

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) [?]. 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). Za rukovanje ograničenjem budžeta primijenjena je stroga kaznena metoda. Ako ukupni trošak odabranog portfelja S ne prelazi budžet, njegova pogodnost je jednaka ukupnom ROI-u. U suprotnom, pogodnost postaje negativna vrijednost proporcionalna iznosu prekoračenja:

$$\text{Fitness}(S) = \begin{cases} \sum_{i \in S} \text{ROI}_i, & \text{ako } \sum_{i \in S} \text{Trošak}_i \leq \text{Budžet} \\ -(\sum_{i \in S} \text{Trošak}_i - \text{Budžet}), & \text{ako } \sum_{i \in S} \text{Trošak}_i > \text{Budžet} \end{cases}$$

Ovakav pristup osigurava da svako valjano rješenje (koje ima pozitivan fitness) uvijek bude ocijenjeno kao bolje od bilo kojeg nevaljanog rješenja (koje ima negativan fitness).

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:

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

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 te Seaborn i Matplotlib [?, ?] 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 ?? te pružaju uvid u dinamiku ponašanja genetskog algoritma.

Tablica 2: Rezultati ablacijske studije za parametre GA

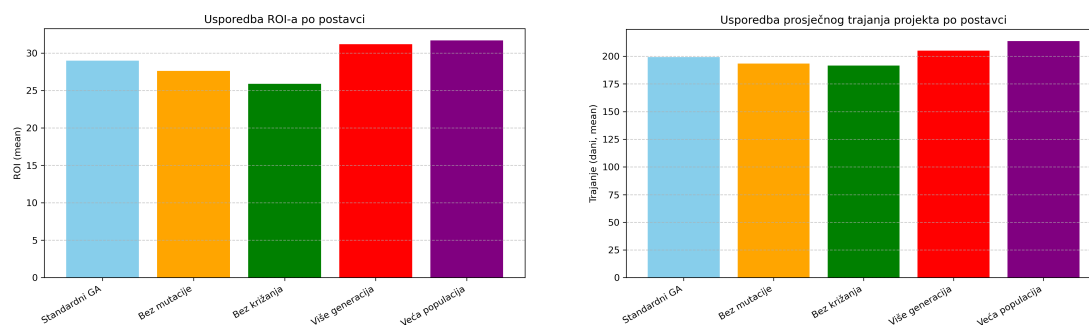
Postavka	ROI_mean	ROI_std	Trajanje_mean	Trajanje_std
Standardni GA	28.985	1.543	199.216	10.691
Bez mutacije	27.627	1.581	193.497	11.364
Bez križanja	25.884	1.865	191.514	9.174
Više generacija	31.183	0.928	205.026	13.649
Veća populacija	31.683	0.720	213.694	5.574

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.

5.4 Eksperiment 2: Usporedna analiza optimizacijskih modela

Na slici ?? prikazan je dijagram toka izvođenja eksperimenta.

Nakon što je u Eksperimentu 1 provedena kalibracija i odabrana "šampionska" konfiguracija genetskog algoritma, u drugoj fazi istraživanja pristupilo se ključnoj usporednoj analizi triju razvijenih modela.

5.4.1 Metodologija

Cilj ovog eksperimenta bio je kvantitativno i kvalitativno usporediti performanse, kvalitetu i stabilnost rješenja dobivenih pomoću tri različita optimizacijska scenarija. Eksperimenti su provedeni prema planu definiranom u Tablici ?. Svaka konfiguracija iz plana testirana je 10 puta (RUNS=10) radi osiguravanja statističke robusnosti zaključaka. Genetski algoritmi (GA (samo ROI) i GA+MC (NSGA-II)) koristili su "šampionsku" konfiguraciju parametara (POP_SIZE = 200, NGEN = 40, itd.) utvrđenu u prethodnom koraku, uz skaliranje parametra NGEN sukladno složenosti problema.

Eksperiment	NUM_ACTIVITIES	BUDGET	Pripada seriji	Napomena
A1	10	1000	A	Osnovna složenost
A2 / B2	50	2500	A, B	Centralni / Referentni eksperiment
A3	100	5000	A	Visoka složenost
B1	50	1500	B	Restriktivan budžet
B3	50	4000	B	Labav budžet

Tablica 3: Plan naprednih eksperimenata

5.4.2 Rezultati

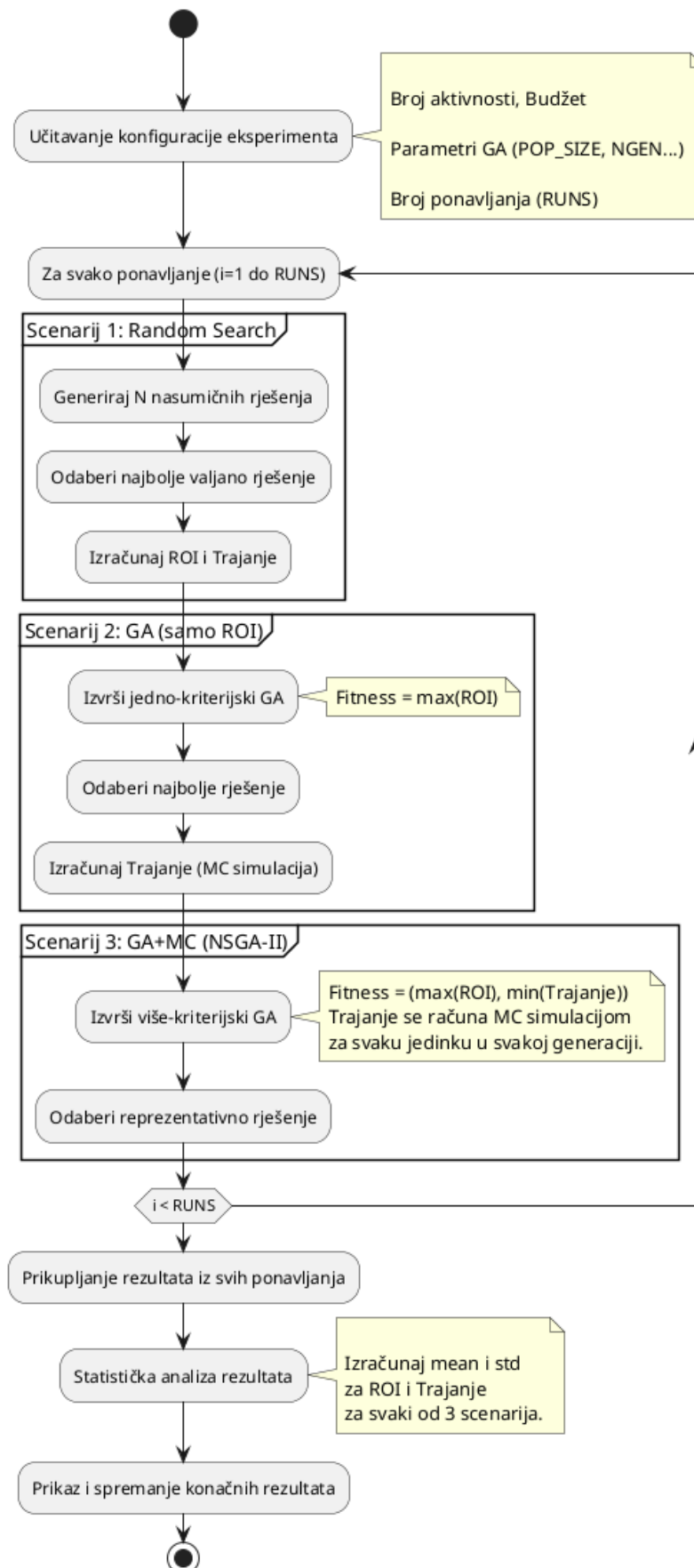
Svi rezultati dobiveni provođenjem Eksperimenta 2 sažeti su u Tablici ?. Ova tablica predstavlja temelj za daljnju diskusiju i donošenje zaključaka.

5.4.3 Diskusija rezultata

Detaljna analiza rezultata provedena je kroz tri tematske cjeline, uz oslanjanje na vizualizacije generirane iz podataka u Tablici ?.

Analiza Skalabilnosti (Serija A)

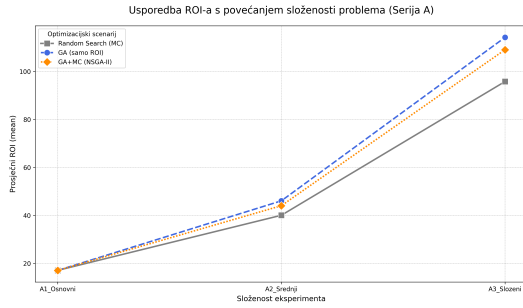
Tok izvođenja Eksperimenta 2: Usporedna studija modela



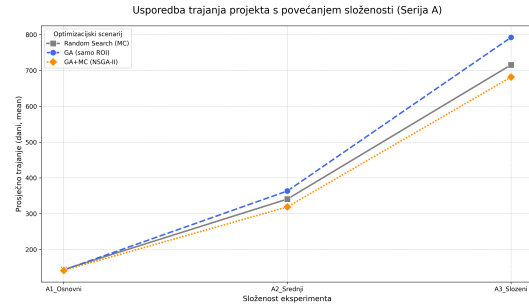
Slika 4: Dijagram toka izvođenja eksperimenta

Eksperiment	Scenarij	ROI_mean	ROI_std	Trajanje_mean	Trajanje_std
A1_Osnovni	Random Search (MC)	17.140	3.55e-15	144.15	10.911
A1_Osnovni	GA (samo ROI)	17.140	3.55e-15	143.10	1.239
A1_Osnovni	GA+MC (NSGA-II)	17.140	3.55e-15	142.00	0.372
A2_Srednji	Random Search (MC)	40.108	0.703	341.02	9.405
A2_Srednji	GA (samo ROI)	46.125	0.412	363.63	7.843
A2_Srednji	GA+MC (NSGA-II)	44.099	0.980	319.21	13.171
A3_Slozeni	Random Search (MC)	95.835	1.468	715.60	10.451
A3_Slozeni	GA (samo ROI)	114.224	0.891	792.30	11.933
A3_Slozeni	GA+MC (NSGA-II)	109.095	2.008	681.56	21.733
B1_Restriktivan	Random Search (MC)	24.120	1.846	197.62	20.181
B1_Restriktivan	GA (samo ROI)	37.976	0.567	253.56	12.386
B1_Restriktivan	GA+MC (NSGA-II)	17.711	17.728	50104.38	49894.619
B3_Labav	Random Search (MC)	71.379	1.114	536.90	16.247
B3_Labav	GA (samo ROI)	79.065	0.518	562.19	9.345
B3_Labav	GA+MC (NSGA-II)	76.949	0.487	526.61	10.602

Tablica 4: Konačni rezultati usporedne analize optimizacijskih modela



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



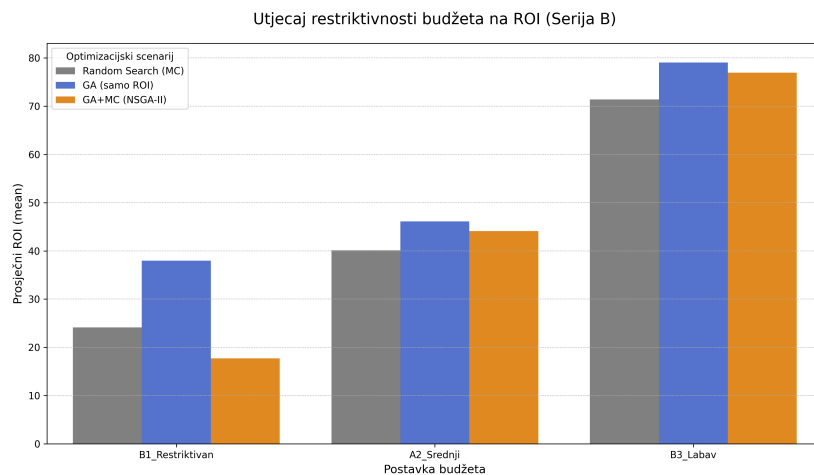
(b) Usporedba prosječnog trajanja projekta za tri različite konfiguracije.

Slika 5: Grafički prikaz rezultata usporednih studija.

Kao što je vidljivo na grafikonima ?? i ??, porast složenosti problema s 10 na 100 aktivnosti drastično utječe na performanse modela. Dok su na osnovnom problemu (A1) sve metode pronašle isti financijski optimum, jaz u ROI_mean vrijednostima eksponencijalno raste u korist genetskih algoritama. U eksperimentu A3, GA (samo ROI) ostvaruje prosječni ROI za preko 18 bodova viši od Random Search metode, što nedvojbeno potvrđuje hipotezu o nužnosti inteligentne pretrage (H1). Istovremeno, analiza trajanja otkriva postojanje kompromisa. Hibridni model GA+MC (NSGA-II) konzistentno identificira rješenja sa značajno nižim prosječnim trajanjem. Na složenom problemu A3, ta razlika iznosi preko 110 dana u usporedbi s GA (samo ROI). Ovo potvrđuje hipotezu H2 – hibridni model uspješno upravlja rizikom, ali uz mjerljivu "cijenu" u vidu nešto nižeg maksimalnog ROI-a.

Analiza Utjecaja Ograničenja (Serija B)

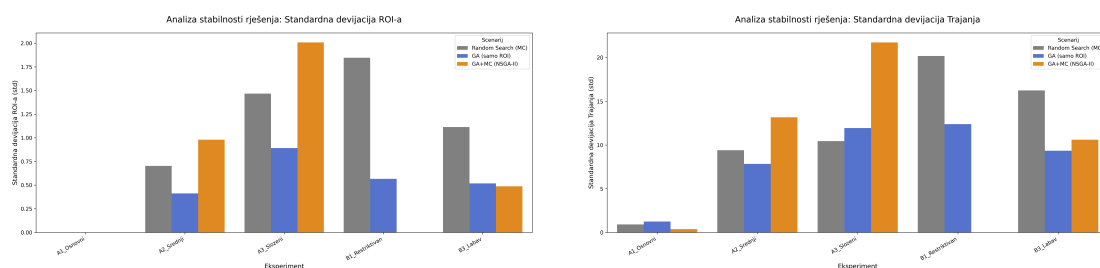
Grafikon ?? ilustrira ponašanje modela pod različitim proračunskim pritiskom. Najvažniji nalaz dolazi iz eksperimenta s restriktivnim budžetom (B1). U tim uvjetima, GA+MC (NSGA-II) pokazuje iznimnu krhkost, ne uspijevajući pronaći valjano rješenje u 50% pokretanja, što rezultira katastrofalnim prosječnim perfor-



Slika 6: Budžet

mansama (Tablica ??, redak 11). S druge strane, jednostavniji GA (samo ROI) pokazuje se vrlo robusnim, uspješno pronalazeći visokoprofitabilna rješenja čak i u vrlo ograničenom prostoru. Ovo ukazuje da složenost više-kriterijske pretrage može biti nedostatak u ekstremno suženim prostorima rješenja. U uvjetima labavog budžeta (B3), svi modeli rade očekivano dobro, a razlike među njima se smanjuju, potvrđujući hipotezu H3.

Analiza Stabilnosti i Pouzdanosti



(a) Stabilnost ROI-a za tri različite konfiguracije.

(b) Stabilnost trajanja projekta za tri različite konfiguracije.

Slika 7: Grafički prikaz rezultata usporednih studija.

Grafikoni ?? i ?? prikazuju standardnu devijaciju kao mjeru konzistentnosti. Izvan scenarija B1 gdje je doživio neuspjeh, GA+MC (NSGA-II) model pokazuje usporedivu ili nižu devijaciju trajanja u odnosu na klasični GA. To implicira da rješenja koja nudi nisu samo u prosjeku brža, već su i pouzdanija, odnosno njihovo procijenjeno trajanje manje varira. Ova predvidljivost je od iznimne važnosti za praktično upravljanje projektima.

5.4.4 Sinteza glavnih zaključaka eksperimenata

- **Random Search (MC):** Koristan kao početna točka i za jednostavne probleme, ali potpuno neadekvatan kao ozbiljan optimizacijski alat za probleme

realne veličine i složenosti.

- **GA (samo ROI):** Izuzetno snažan i robustan "profitni maksimizator". Najbolji je izbor u situacijama gdje je financijska dobit jedini i isključivi kriterij, te pokazuje veliku otpornost u uvjetima strogih ograničenja.
- **GA+MC (NSGA-II):** Sofisticirani "upravitelj rizikom". Njegova najveća vrijednost je u pružanju strateških opcija koje balansiraju profit i rizik (trajanje). Superioran je u standardnim i složenim uvjetima, ali njegova složenost ga čini osjetljivim i nepouzdanim u okruženjima s ekstremno restriktivnim ograničenjima.

Konačan izbor modela stoga ovisi o strateškim prioritetima projektnog ureda. Za maksimalan profit, klasični GA je pobjednik. Za uravnoteženo i rizikom informirano donošenje odluka, hibridni GA+MC je superioran, uz nužan oprez pri primjeni u vrlo ograničenim uvjetima.

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 [?, ?], omogućili smo efikasno pretraživanje velikog prostora rješenja, dok je Monte Carlo simulacija služila za kvantitativno modeliranje rizika i nesigurnosti [?, ?], 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 [?].
- 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 [?], 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] Python Software Foundation. Python language reference, version 3.x. <https://www.python.org/>, 2024. Accessed: 2025-07-31.
- [21] 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.
- [22] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

Popis slika

Popis tablica