

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: 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 razvijen je i evaluiran hibridni optimizacijski pristup temeljen na genetskim algoritmima (GA) i Monte Carlo (MC) simulaciji. Kroz dvo-fazni eksperimentalni dizajn, provedena je sustavna usporedba tri modela: nasumične pretrage, jedno-kriterijskog GA usmjerenog isključivo na povrat na investiciju (ROI), te više-kriterijskog GA+MC modela (NSGA-II) koji istovremeno optimizira ROI i rizik trajanja projekta. Rezultati dobiveni na sintetičkim podacima različite složenosti i restriktivnosti pokazuju da, iako jedno-kriterijski GA najučinkovitije maksimizira profit, hibridni GA+MC model uspješno generira Paretov front rješenja koja nude optimalan kompromis između profitabilnosti i trajanja. Nadalje, istraživanje je otkrilo ključan nalaz: pod ekstremno restriktivnim budžetom, robusnost jednostavnijeg, jedno-kriterijskog GA nadmašuje onu složenijeg, više-kriterijskog modela. Rad zaključuje da ne postoji univerzalno superioran model, već da optimalan izbor ovisi o strateškim prioritetima – maksimizaciji profita naspram uravnoteženog upravljanja rizikom.

**Ključne riječi:** projektno upravljanje, genetski algoritam, Monte Carlo simulacija, više-kriterijska optimizacija, upravljanje rizikom, Paretov front, NSGA-II

## Abstract

Project management often involves complex decisions regarding the allocation of activities and resources, especially under conditions of uncertainty and constraints. Traditional methods such as PERT and CPM frequently fail to capture the stochastic nature of real-world projects. This thesis develops and evaluates a hybrid optimization approach based on genetic algorithms (GA) and Monte Carlo (MC) simulation. Through a two-phase experimental design, a systematic comparison of three models was conducted: random search, a single-objective GA focused solely on return on investment (ROI), and a multi-objective GA+MC model (NSGA-II) that simultaneously optimizes ROI and project duration risk. The results, obtained from synthetic data of varying complexity and restrictiveness, show that while the single-objective GA is most effective at maximizing profit, the hybrid GA+MC model successfully generates a Pareto front of solutions offering an optimal trade-off between profitability and duration. Furthermore, the research revealed a key finding: under extremely restrictive budget constraints, the robustness of the simpler, single-objective GA surpasses that of the more complex, multi-objective model. The thesis concludes that there is no universally superior model; rather, the optimal choice depends on strategic priorities—maximizing profit versus balanced risk management.

**Keywords:** project management, genetic algorithm, Monte Carlo simulation, multi-objective optimization, risk management, Pareto front, NSGA-II

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Motivacija i značaj istraživanja . . . . .	1
1.2	Izazov nesigurnosti u projektnom upravljanju . . . . .	2
1.3	Ciljevi, doprinos i struktura rada . . . . .	2
<b>2</b>	<b>Teorijske osnove i srodni radovi</b>	<b>4</b>
2.1	Problem odabira portfelja kao Knapsack problem . . . . .	4
2.2	Genetski algoritmi kao metaheuristika za pretraživanje . . . . .	4
2.2.1	Više-kriterijski genetski algoritmi i NSGA-II . . . . .	5
2.3	Kvantifikacija rizika pomoću Monte Carlo simulacije . . . . .	6
2.4	Modeliranje nesigurnosti trajanja pomoću trotočkovne procjene . . . . .	6
<b>3</b>	<b>Model problema</b>	<b>8</b>
3.1	Definicija skupa aktivnosti i varijabli odluke . . . . .	8
3.2	Resursna ograničenja modela . . . . .	8
3.3	Specifikacija optimizacijskih ciljeva . . . . .	8
3.4	Konceptualni prikaz modela . . . . .	9
<b>4</b>	<b>Arhitektura sustava i implementacija</b>	<b>10</b>
4.1	Korištene tehnologije i biblioteke . . . . .	10
4.2	Arhitektura eksperimentalnog okvira . . . . .	10
4.3	Modeliranje nesigurnosti: Monte Carlo simulacija . . . . .	13
4.4	Optimizacijski pristup: Genetski algoritam . . . . .	14
4.5	Vizualizacija i obrada rezultata . . . . .	15
<b>5</b>	<b>Eksperimenti i analiza rezultata</b>	<b>17</b>
5.1	Postavke okruženja i testni podaci . . . . .	17
5.2	Eksperimentalni dizajn . . . . .	17
5.2.1	Istraživačke hipoteze . . . . .	18
5.3	Eksperiment 1: Analiza parametara i kalibracija genetskog algoritma . . . . .	18
5.4	Eksperiment 2: Usporedna analiza optimizacijskih modela . . . . .	19
5.4.1	Rezultati . . . . .	19
5.4.2	Diskusija rezultata . . . . .	20
5.5	Sinteza glavnih zaključaka eksperimenata . . . . .	22
<b>6</b>	<b>Zaključak</b>	<b>23</b>
6.1	Glavni nalazi i odgovori na istraživačka pitanja . . . . .	23
6.2	Doprinos rada . . . . .	23
6.3	Ograničenja rada . . . . .	24
6.4	Preporuke za budući rad . . . . .	24
6.5	Završna riječ . . . . .	24
	<b>Literatura</b>	<b>26</b>
	<b>Popis slika</b>	<b>28</b>



# 1 Uvod

U suvremenom projektnom upravljanju, alokacija ograničenih resursa na niz konkurentskih aktivnosti predstavlja jedan od fundamentalnih izazova. Ovaj problem optimizacije može se elegantno modelirati kroz klasični *problem ruksaka* (*Knapsack Problem*), gdje je cilj odabrati podskup stavki na način koji maksimizira ukupnu vrijednost, a da se pritom ne prekorači zadani kapacitet [1]. U domeni projekata, to se preslikava na odabir portfelja aktivnosti koji će generirati najveći mogući povrat na investiciju (ROI), uz strogo poštivanje proračunskih, vremenskih i drugih resursnih ograničenja.

Iako tradicionalne metode poput PERT-a (*Program Evaluation and Review Technique*) i CPM-a (*Critical Path Method*) [2,3] pružaju nezamjenjiv okvir za planiranje i praćenje, njihova primjena nailazi na poteškoće u dinamičnim okruženjima. One su često determinističke prirode i teško se nose s dva ključna izazova: kombinatornom eksplozijom mogućih rješenja i inherentnom nesigurnošću koja prati procjene trajanja i troškova.

Pregled literature ukazuje na to da su genetski algoritmi (GA) iznimno učinkoviti u rješavanju NP-teških optimizacijskih problema [4,5], dok Monte Carlo (MC) simulacije predstavljaju zlatni standard za modeliranje i kvantifikaciju rizika [6,7]. Unatoč tome, primjetan je nedostatak radova koji sustavno istražuju sinergiju ovih dviju moćnih tehnika. Upravo se u tom prostoru pozicionira ovaj rad, s ciljem popunjavanja uočene praznine kroz razvoj i rigoroznu evaluaciju hibridnog GA-MC modela. Fokus nije samo na kombinaciji, već na stvaranju sustavnog eksperimentalnog okvira za kalibraciju i usporednu analizu takvih modela, kako bi se precizno kvantificirao utjecaj ove sinergije na robusnost i kvalitetu konačnih rješenja.

## 1.1 Motivacija i značaj istraživanja

Motivacija za ovo istraživanje proizlazi iz fundamentalne razlike između projektnog plana i stvarne izvedbe. U praksi, projekti se rijetko odvijaju točno prema početnim predviđanjima [3]. Fluktuacije u dostupnosti resursa, nepredviđeni događaji, izmjene u zahtjevima dionika te inherentna nepreciznost početnih procjena česti su uzroci odstupanja, što klasične metode planiranja čini nedovoljno fleksibilnima [8].

Stoga se javlja potreba za pristupima koji nadilaze puku alokaciju resursa i ulaze u domenu strateškog upravljanja nesigurnošću. Potrebne su metode koje ne samo da pronalaze matematički optimalan raspored, već vrednuju rješenja i prema njihovoj otpornosti na nepredviđene okolnosti. Ovdje komplementarna priroda genetskih algoritama i Monte Carlo simulacija dolazi do punog izražaja. Genetski algoritmi pružaju mehanizam za efikasno pretraživanje astronomski velikog prostora mogućih portfelja [5], dok Monte Carlo simulacije služe kao "simulator stvarnosti", testirajući svako potencijalno rješenje u tisućama mogućih budućnosti kako bi se procijenila njegova stvarna varijabilnost i rizik [7].

Kombiniranjem ovih metoda, ovaj rad prelazi s tradicionalnog pitanja "Koji je najbolji plan?" na puno značajnije pitanje: "Koji je najrobusniji plan koji nudi najbolji kompromis između profita i rizika?".

## 1.2 Izazov nesigurnosti u projektnom upravljanju

Rizici u projektnom upravljanju obuhvaćaju sve događaje ili uvjete koji, ako se pojave, mogu negativno utjecati na ciljeve projekta [9]. Oni mogu biti tehničke, organizacijske, financijske ili tržišne prirode, a često su međusobno povezani [8]. Nesigurnosti, s druge strane, proizlaze iz nepotpunih informacija, promjenjivih uvjeta i fundamentalne nemogućnosti preciznog predviđanja budućih događaja [10]. Učinkovito upravljanje projektima stoga zahtijeva sustavan pristup identifikaciji, procjeni i razvoju strategija odgovora na rizike i nesigurnosti [9].

U domeni kvantitativne analize, Monte Carlo simulacije predstavljaju jedan od najmoćnijih pristupa [11]. Metoda se temelji na generiranju velikog broja mogućih scenarija temeljenih na specificiranim distribucijama vjerojatnosti za neizvjesne ulazne parametre. Analizom distribucije dobivenih ishoda, moguće je s visokom pouzdanošću procijeniti, primjerice, vjerojatnost završetka projekta unutar zadanog roka ili budžeta [7].

## 1.3 Ciljevi, doprinos i struktura rada

Primarni cilj ovog diplomskog rada jest razviti i kroz sustavne eksperimente evaluirati hibridni GA-MC model za optimizaciju portfelja projektnih aktivnosti. Svrha modela je pružiti podršku u donošenju odluka koje su informirane ne samo o potencijalnoj dobiti, već i o pripadajućem riziku. Doprinos rada ostvaren je na teorijskoj, metodološkoj i praktičnoj razini. Na teorijskoj razini, rad povezuje koncepte iz evolucijskog računarstva s kvantitativnim metodama upravljanja rizikom u jedinstven okvir primijenjen na projektno upravljanje. S metodološkog stajališta, razvijen je i validiran cjelovit dvo-fazni proces za evaluaciju složenih hibridnih modela, koji uključuje fazu kalibracije i fazu usporedne analize. Konačno, praktični doprinos očituje se u demonstraciji primjenjivosti pristupa na sintetičkim podacima, čime se dobiva uvid u performanse i ograničenja različitih optimizacijskih strategija.

Kako bi se navedeni ciljevi ostvarili, istraživanje je vođeno sljedećim ključnim pitanjima:

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 integracija Monte Carlo simulacije u proces evaluacije može poboljšati robusnost rješenja dobivenih genetskim algoritmom?
3. Kakav je utjecaj kombiniranog pristupa na stabilnost i pouzdanost optimizacije u uvjetima neizvjesnosti?

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].

Ostatak rada organiziran je kako slijedi. **Drugo poglavlje** pruža pregled relevantne literature i teorijskih osnova, uključujući problem ruksaka, genetske algoritme i metode modeliranja nesigurnosti. **Treće poglavlje** detaljno opisuje matematički i konceptualni model problema koji se rješava. **Četvrto poglavlje** ulazi u detalje implementacije, opisujući arhitekturu razvijenog softverskog sustava i korištene programske biblioteke. **Peto poglavlje** čini srž rada, prikazujući dvo-fazni eksperimentalni dizajn, analizu dobivenih rezultata i detaljnu diskusiju o performansama testiranih modela. Konačno, **šesto poglavlje** donosi sintezu cjelokupnog rada, sažima ključne doprinose, osvrće se na ograničenja provedenog istraživanja i nudi preporuke za budući rad.



## 2 Teorijske osnove i srodni radovi

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 Problem odabira portfelja kao Knapsack problem

Problem ruksaka (engl. *Knapsack Problem*) jedan je od najpoznatijih problema kombinatorne optimizacije, svrstan u klasu NP-teških problema, što znači da ne postoji poznati algoritam koji bi ga mogao riješiti u polinomnom vremenu za sve slučajeve [1, 12]. U svojoj osnovnoj, jednodimenzionalnoj verziji (0/1 Knapsack Problem), cilj je odabrati podskup objekata iz danog skupa, od kojih svaki ima definiranu težinu i vrijednost. Optimizacijski zadatak je maksimizirati ukupnu vrijednost odabranih objekata, pod uvjetom da njihova ukupna težina ne prelazi unaprijed 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.

Ova elegantna formulacija čini ga moćnim alatom za modeliranje problema alokacije resursa u stvarnom svijetu. U kontekstu upravljanja projektima, problem odabira portfelja aktivnosti prirodno poprima oblik složenije, višedimenzionalne varijante poznate kao *Multi-Dimensional Knapsack Problem* (MDKP). U ovom modelu, projektne aktivnosti predstavljaju "objekte" koje želimo staviti u ruksak, a njihova vrijednost je povrat na investiciju (ROI) ili neki drugi pokazatelj strateške važnosti. "Težina" aktivnosti nije više jedna dimenzija, već vektor koji opisuje potrošnju različitih resursa – budžeta, radnih sati, specifične opreme, itd. Kapacitet ruksaka tada odgovara ukupnoj raspoloživosti svakog od tih resursa. MDKP stoga pruža robustan matematički okvir za rješavanje središnjeg problema ovog rada: kako optimalno raspodijeliti ograničene, višestruke resurse među konkurentskim projektnim aktivnostima.

### 2.2 Genetski algoritmi kao metaheuristika za pretraživanje

Genetski algoritmi (GA) su moćne metaheurističke optimizacijske metode inspirirane procesima prirodne selekcije i evolucije [4, 5]. Pripadaju široj klasi evolucijskih algoritama i iznimno su učinkoviti u rješavanju složenih optimizacijskih problema s velikim, diskretnim ili nelinearnim prostorima rješenja, posebno onih NP-teških, za koje klasične metode nisu praktične [13, 14].

Umjesto da pretražuju prostor rješenja s jednom točkom, genetski algoritmi operiraju nad cijelom populacijom potencijalnih rješenja (kromosoma). Taj proces započinje generiranjem početnog, najčešće nasumičnog skupa rješenja. Zatim ulazi u iterativnu petlju u kojoj se simulira evolucija kroz generacije. U svakoj generaciji, svakom se rješenju (jedinki) dodjeljuje vrijednost pogodnosti (fitness) koja kvantificira njegovu kvalitetu prema definiranoj ciljnoj funkciji. Ovisno o cilju, funkcija pogodnosti može biti jedno-kriterijska (npr. maksimizacija ROI-a) ili, kao u naprednijem modelu ovog rada, više-kriterijska, gdje se istovremeno vrednuje više, često suprotstavljenih, ciljeva. Nakon evaluacije, provodi se selekcija, gdje jedinke s boljim fitnessom imaju veću vjerojatnost da budu odabrane kao "roditelji". Ti roditelji se zatim rekombiniraju pomoću operatora križanja (crossover), stvarajući "potomke" koji nasljeđuju i kombiniraju njihove karakteristike. Kako bi se održala raznolikost u populaciji i izbjeglo zaglavljivanje u lokalnim optimumima, na potomke se primjenjuje operator mutacije, koji unosi male, nasumične promjene. Na kraju, nova generacija potomaka zamjenjuje staru, i cijeli se proces ponavlja dok se ne zadovolji unaprijed definirani kriterij zaustavljanja, poput maksimalnog broja generacija.

### 2.2.1 Više-kriterijski genetski algoritmi i NSGA-II

Standardni genetski algoritmi su dizajnirani za probleme s jednim ciljem (jedno-kriterijska optimizacija), gdje je lako usporediti dva rješenja na temelju jedne vrijednosti pogodnosti. Međutim, mnogi stvarni problemi, uključujući i odabir projektnog portfelja, inherentno su više-kriterijski – zahtijevaju istovremeno zadovoljenje više, često suprotstavljenih, ciljeva (npr. maksimizirati profit, a minimizirati rizik).

Za rješavanje takvih problema razvijeni su više-kriterijski evolucijski algoritmi (engl. *Multi-Objective Evolutionary Algorithms*, MOEA). Njihov cilj nije pronaći jedno jedino "najbolje" rješenje, već skup rješenja koji predstavlja optimalan kompromis, poznat kao **Pareto front**. Ključni koncepti na kojima se temelje su:

- **Dominacija:** Rješenje A "dominira" rješenje B ako je bolje ili jednako od B po svim kriterijima, a strogo bolje po barem jednom kriteriju.
- **Pareto front:** Skup svih rješenja u populaciji koja nisu dominirana ni od jednog drugog rješenja. To su najbolja kompromisna rješenja koja algoritam može ponuditi.

Jedan od najpoznatijih i najčešće korištenih algoritama u ovoj domeni je **NSGA-II** (*Nondominated Sorting Genetic Algorithm II*) [15]. Njegova efikasnost leži u dva ključna mehanizma:

1. **Brzo sortiranje po nedominaciji (Fast non-dominated sorting):** U svakoj generaciji, algoritam rangira populaciju u slojeve (frontove) na temelju dominacije. Rješenja u prvom frontu su najbolja, zatim slijede rješenja u drugom frontu, i tako dalje. Ovim se osigurava da elitna, ne-dominirana rješenja imaju prednost pri selekciji.

2. **Procjena gustoće naseljenosti (Crowding distance):** Kako bi se unutar istog fronta osigurala raznolikost i spriječilo grupiranje rješenja u samo jednom dijelu prostora, NSGA-II koristi mjeru "napučenosti". Prednost se daje rješenjima koja se nalaze u rjeđe naseljenim dijelovima Paretovog fronta, čime se dobiva širok i dobro raspoređen skup kompromisnih opcija.

Zbog svoje robusnosti i efikasnosti u pronalaženju dobro raspoređenog Paretovog fronta, NSGA-II je odabran kao temelj za više-kriterijski hibridni model u ovom radu.

## 2.3 Kvantifikacija rizika pomoću Monte Carlo simulacije

Monte Carlo simulacija (MCS) je računaska metoda koja koristi nasumično uzorkovanje za numeričko modeliranje i analizu složenih, stohastičkih sustava [7]. Njena primarna prednost leži u sposobnosti da pruži uvid u distribuciju mogućih ishoda nekog procesa, umjesto da daje samo jednu determinističku procjenu, što je čini nezamjenjivim alatom za kvantitativnu analizu rizika [11]. U projektnom upravljanju, MCS se široko primjenjuje za procjenu vjerojatnih ishoda trajanja i troškova projekata, uzimajući u obzir inherentnu varijabilnost svake pojedine aktivnosti [16, 17].

Proces Monte Carlo simulacije odvija se u nekoliko koraka. Prvo se identificiraju ključne ulazne varijable čije su vrijednosti neizvjesne, poput trajanja pojedinih aktivnosti. Za svaku takvu varijablu odabire se distribucija vjerojatnosti koja najbolje opisuje raspon i vjerojatnost njenih mogućih vrijednosti. Zatim, algoritam iterativno provodi tisuće "eksperimenata": u svakoj iteraciji, za svaku nesigurnu varijablu generira se jedna nasumična vrijednost iz njene definirane distribucije, te se na temelju tih vrijednosti izračunava ishod modela (npr. ukupno trajanje projekta). Ponavljanjem ovog procesa velik broj puta, dobiva se empirijska distribucija mogućih ishoda, iz koje se mogu iščitati ključni statistički pokazatelji poput očekivane vrijednosti, standardne devijacije i vjerojatnosti prekoračenja određenih pragova.

## 2.4 Modeliranje nesigurnosti trajanja pomoću trotočkovne procjene

Metodologija PERT (*Program Evaluation and Review Technique*) uvela je u praksu korištenje tri vremenske procjene za aktivnosti s neizvjesnim trajanjem, što je danas standard u upravljanju projektnim rizikom [2]. Ove tri točke su:  $T_o$  (optimistična procjena),  $T_m$  (najvjerojatnija procjena) i  $T_p$  (pesimistična procjena).

Dok tradicionalna PERT metoda koristi ove tri točke za izračun parametara Beta distribucije, u modernoj praksi, a posebno u Monte Carlo simulacijama, često se koristi Trokutasta distribucija zbog svoje jednostavnosti i intuitivnosti [18].

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 koje čine ulaz u model, primijenjena resursna ograničenja te jedno-kriterijski i više-kriterijski ciljevi optimizacije koji su korišteni u kasnijoj eksperimentalnoj evaluaciji.

#### 3.1 Definicija skupa aktivnosti i varijabli odluke

Problem se definira kao odabir optimalnog podskupa (portfelja) aktivnosti iz većeg, unaprijed definiranog skupa dostupnih aktivnosti, što je čest izazov u projektnom menadžmentu [3, 8]. Neka je  $A = \{a_1, a_2, \dots, a_n\}$  skup od  $n$  dostupnih projektnih aktivnosti. Svaka aktivnost  $a_i \in A$  opisana je s tri ključna atributa. Prvi je trošak ( $c_i$ ), koji predstavlja količinu budžeta potrebnu za izvođenje aktivnosti. Drugi atribut je vrijednost ( $v_i$ ), definirana kao povrat na investiciju (ROI) koji se ostvaruje uspješnim završetkom. Treći, ključni atribut je nesigurnost trajanja, koja se ne modelira kao jedna deterministička vrijednost, već kao stohastička varijabla opisana s tri točke procjene: optimističnom ( $T_o$ ), najvjerojatnijom ( $T_m$ ) i pesimističnom ( $T_p$ ), koje služe kao parametri za Trokutastu distribuciju u Monte Carlo simulaciji.

S obzirom na navedene attribute, temeljni zadatak optimizacije je definirati binarni vektor odluke  $x = (x_1, x_2, \dots, x_n)$ , gdje  $x_i \in \{0, 1\}$ . Vrijednost  $x_i = 1$  označava da je aktivnost  $a_i$  odabrana za uključivanje u portfelj, dok  $x_i = 0$  označava da se aktivnost ne izvodi.

#### 3.2 Resursna ograničenja modela

Iako realni projekti mogu imati višestruka ograničenja, u ovom modelu implementirano je ključno i najčešće ograničenje u upravljanju portfeljem: ograničenje budžeta ( $B_{max}$ ). Prema ovom pravilu, ukupni zbroj troškova svih odabranih aktivnosti ne smije prelaziti ukupno raspoloživi budžet. Formalno, ovo ograničenje odgovara klasičnoj formulaciji problema ruksaka (*Knapsack Problem*) [1] i matematički se izražava kao:

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

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 [10].

#### 3.3 Specifikacija optimizacijskih ciljeva

U skladu s eksperimentalnim dizajnom, definirana su i analizirana dva različita optimizacijska cilja, što odgovara dvama glavnim testiranim scenarijima.

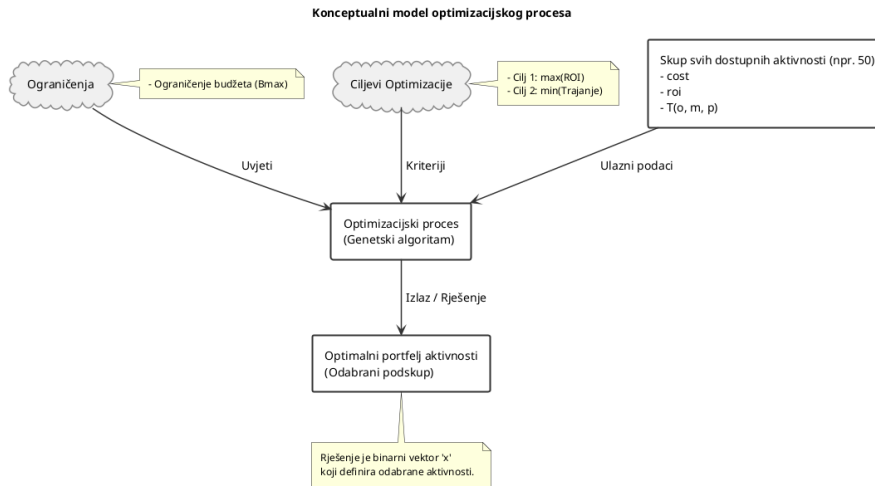
Prvi, jedno-kriterijski cilj, usmjeren je isključivo na maksimizaciju financijske dobiti. Ovaj pristup odgovara klasičnom GA (samo ROI) modelu, a njegova ciljna funkcija je maksimizacija ukupnog zbroja ROI vrijednosti odabranih aktivnosti, uz poštivanje ograničenja budžeta. Ovakav tip optimizacijskog cilja čest je u primjeni genetskih algoritama na probleme alokacije resursa [5], a formalno se zapisuje kao:

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

S druge strane, više-kriterijski cilj predstavlja srž ovog istraživanja i odgovara naprednom hibridnom GA+MC (NSGA-II) modelu. U ovom pristupu istovremeno se optimiziraju dva suprotstavljena cilja: (1) maksimizacija ukupnog ROI-a i (2) minimizacija procijenjenog trajanja portfelja. Budući da poboljšanje jednog cilja često degradira drugi, ovaj pristup ne traži jedno jedino "najbolje" rješenje. Umjesto toga, cilj je pronaći skup optimalnih kompromisnih rješenja (tzv. Pareto front), za što je korišten renomirani NSGA-II algoritam [15].

### 3.4 Konceptualni prikaz modela

Konceptualni model problema, koji objedinjuje sve prethodno opisane elemente, prikazan je na Slici 1. Dijagram ilustrira proces u kojem optimizacijski algoritam, kao središnji mehanizam, uzima skup svih dostupnih aktivnosti kao ulaz. Vođen definiranim ciljevima (maksimizacija ROI-a i/ili minimizacija trajanja) i ograničen zadanom budžetom, algoritam pretražuje prostor mogućih kombinacija kako bi kao izlaz generirao optimalni portfelj – podskup aktivnosti koji najbolje zadovoljava postavljene kriterije.



Slika 1: Konceptualni model problema optimizacije portfelja projektnih aktivnosti.

Ovaj formalizirani model omogućuje jasnu matematičku i vizualnu reprezentaciju optimizacijskog zadatka. Precizna definicija ulaza, ograničenja i ciljeva ključna je za sustavnu primjenu optimizacijskih metoda poput genetskih algoritama [14] u kombinaciji s Monte Carlo simulacijama [7], kako bi se dobila rješenja visoke kvalitete koja su istovremeno i robusna na prisutne nesigurnosti.

## 4 Arhitektura sustava i implementacija

Nakon definiranja teorijskih osnova i formalizacije problema, ovo poglavlje detaljno opisuje praktičnu realizaciju razvijenog optimizacijskog modela. Sustav je implementiran u programskom jeziku **Python (verzija 3.x)**, odabranom zbog čitljivosti, bogatog ekosustava biblioteka i široke primjene u znanstvenom računarstvu [19]. Python omogućuje brzu izradu prototipa, jednostavnu integraciju modula te učinkovitu obradu i vizualizaciju podataka.

### 4.1 Korištene tehnologije i biblioteke

Za izradu sustava korišten je niz standardnih biblioteka iz Python ekosustava za znanstveno računarstvo. Biblioteka **DEAP** [20] korištena je kao temeljni okvir za implementaciju genetskog algoritma. Za numeričke operacije i statističku obradu korišten je **NumPy** [21], dok je za manipulaciju i spremanje tabličnih podataka korišten **Pandas** [22]. Vizualizacija rezultata provedena je pomoću biblioteka **Seaborn** [23] i **Matplotlib** [24]. Modeliranje nesigurnosti pomoću Trokutaste distribucije implementirano je korištenjem standardne Python biblioteke Random. Detaljan pregled dan je u Tablici 1.

Tablica 1: Korištene biblioteke u implementaciji

Biblioteka	Namjena i citat
Python	Osnovni programski jezik za cjelokupnu implementaciju. [19]
DEAP	Okvir za razvoj i provedbu evolucijskih algoritama. [20]
NumPy	Numeričke operacije i statistička obrada nizova podataka. [21]
Pandas	Učitavanje, obrada i spremanje tabličnih podataka s rezultatima. [22]
Seaborn	Kreiranje naprednih statističkih vizualizacija (stupčasti, linijski i raspršeni grafikoni). [23]
Matplotlib	Osnovna biblioteka za crtanje na koju se oslanja Seaborn. [24]
Random	Standardna Python biblioteka korištena za generiranje slučajnih brojeva i uzorkovanje iz Trokutaste distribucije.

### 4.2 Arhitektura eksperimentalnog okvira

Sustav razvijen za potrebe ovog rada nije zamišljen kao monolitna skripta, već kao cjeloviti i modularni eksperimentalni okvir. Takva arhitektura, prikazana na Slici ??, omogućuje sustavnu analizu, kalibraciju i usporedbu optimizacijskih metodologija. Okvir se sastoji od dva glavna analitička modula koji slijede dvofazni pristup istraživanju, te jednog pomoćnog modula za obradu i prikaz rezultata.

Prvi modul, **Modul za analizu i kalibraciju genetskog algoritma**, predstavlja temelj istraživanja i odgovara na pitanje: *“Kako optimalno konfigurirati*

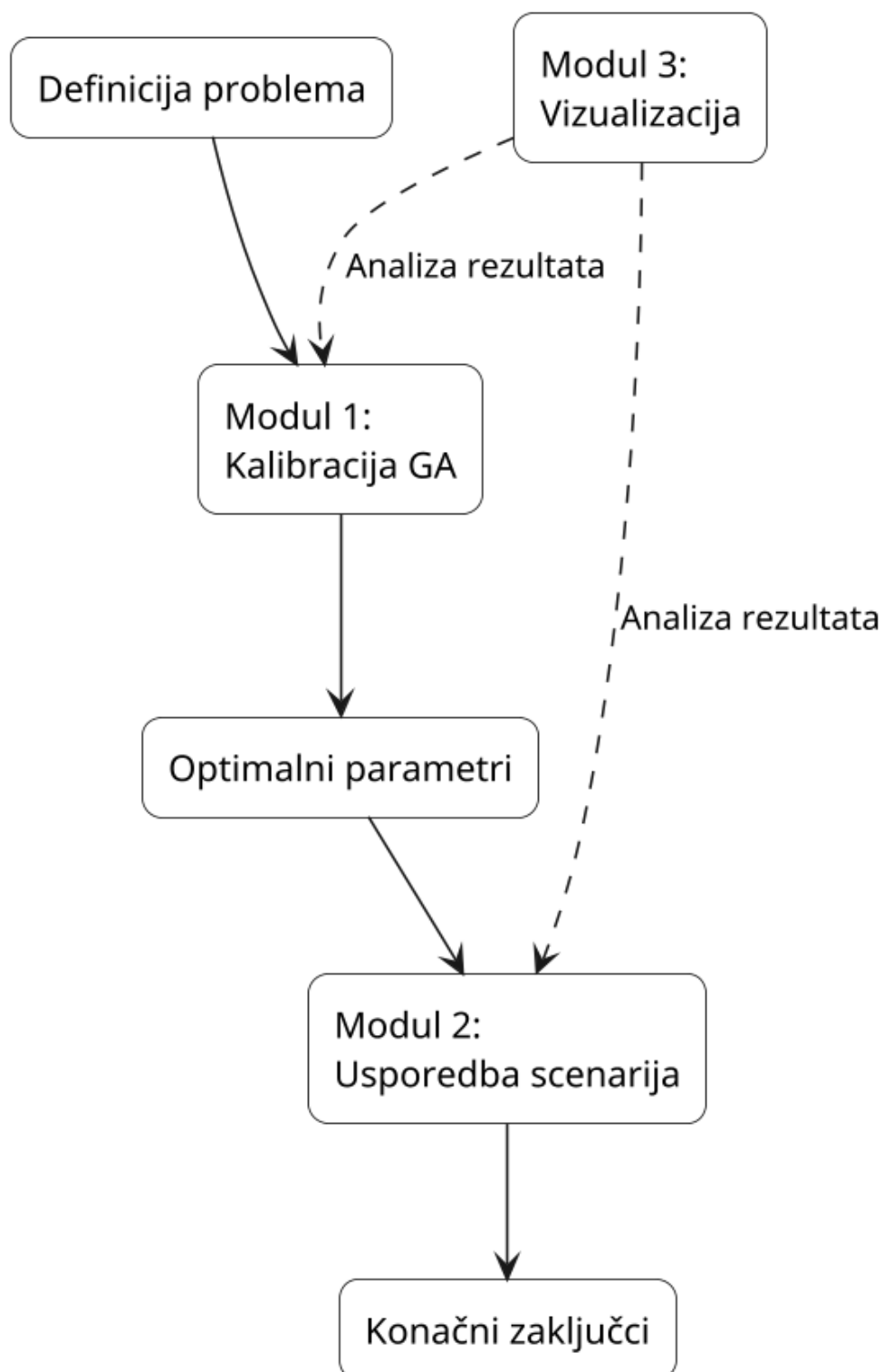
*genetski algoritam za rješavanje zadanog problema?*". Njegova primarna svrha je provođenje detaljne ablacijske studije kojom se ispituje utjecaj svakog ključnog parametra na performanse. Kroz višestruka pokretanja različitih konfiguracija (standardni GA, bez križanja, bez mutacije, s povećanim brojem generacija, s većom populacijom), izračunavanjem metrika performansi, uključujući prosječnu vrijednost (mean) i standardnu devijaciju (std) za ROI i procijenjeno trajanje, ovaj modul kao izlaz generira "šampionsku" konfiguraciju – skup optimalnih parametara koji osiguravaju najbolje performanse i koji se koriste u daljnjoj analizi.

Drugi, ključni modul je **Modul za usporednu analizu optimizacijskih scenarija**. On čini srž diplomskog rada i koristi "šampionsku" konfiguraciju za provođenje konačne, statistički robusne usporedbe triju različitih pristupa rješavanju problema: osnovnog modela nasumične pretrage, klasičnog genetskog algoritma usmjerenog isključivo na ROI, te hibridnog GA+MC modela (NSGA-II) koji provodi više-kriterijsku optimizaciju koja istovremeno maksimizira ROI i minimizira rizik trajanja procijenjen Monte Carlo simulacijom. Izlaz ovog modula je konačna tablica s usporednim rezultatima performansi (ROI, trajanje) i stabilnosti (standardna devijacija) za svaki od triju scenarija.

Treći, **Modul za obradu i vizualizaciju rezultata**, služi kao pomoćni alat za interpretaciju podataka dobivenih iz prva dva modula, generirajući pregledne tablice i grafičke prikaze kao što su stupčasti dijagrami za usporedbu prosječnih vrijednosti ili 2D raspršeni dijagrami (scatter plots) za prikaz Paretovog fronta dobivenog iz NSGA-II algoritma generirane iz rezultata spremnjenih u CSV formatu.



## 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  $i$ .

Implementacija se temelji na `monte_carlo_eval_duration` funkciji.

```
1 def monte_carlo_eval_duration(individual, activities,
2   config):
3     selected = [act for i, act in enumerate(activities) if
4                 individual[i] == 1]
5     if not selected:
6         return 0.0
7     durations = [
8         sum(
9             random.triangular(act["optimistic"], act["
10                realistic"], act["pessimistic"])
11             for act in selected
12         )
13     ]
14     return np.mean(durations)
```

Listing 1: Funkcija za Monte Carlo procjenu trajanja

**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) [20]. S obzirom na prirodu problema odabira podskupa aktivnosti, korištena je **binarna reprezentacija**, gdje svaka jedinka (kromosom) predstavlja jedan portfelj projekata u obliku binarnog niza. Vrijednost '1' na poziciji  $i$  označava da je  $i$ -ta aktivnost odabrana, dok '0' označava da nije.

**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).** Ključni dio implementacije su funkcije pogodnosti. Ovisno o scenariju, korištene su dvije različite funkcije. Za jedno-kriterijsku optimizaciju (GA samo ROI), implementirana je funkcija koja maksimizira ROI i primjenjuje strogu kaznenu metodu za rješenja koja prekoračuju budžet, osiguravajući da su sva nevaljana rješenja lošija od bilo kojeg valjanog. Za hibridni scenarij (GA+MC), implementirana je više-kriterijska funkcija pogodnosti koja predstavlja srž ovog rada. Kao što je prikazano u Isječku koda 2, ova funkcija unutar jedne evaluacije objedinjuje deterministički izračun (ukupni trošak i ROI) i poziv stohastičke Monte Carlo simulacije za procjenu rizika (očekivano trajanje). Vraća tuple s dvije vrijednosti, omogućujući NSGA-II algoritmu da istovremeno optimizira oba cilja. 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:

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

```

1 def multi_objective_fitness(individual, activities, config)
  :
2     total_cost, total_roi = calculate_metrics(individual,
        activities)
3     if total_cost > config["BUDGET"]:
4         return 0, 99999
5     avg_duration = monte_carlo_eval_duration(individual,
        activities, config)
6     return total_roi, avg_duration

```

Listing 2: Više-kriterijska funkcija pogodnosti

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:

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

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

**Genetski operatori.** Za evoluciju populacije korišteni su sljedeći standardni genetski operatori za binarnu reprezentaciju koje pruža DEAP. Za **Selekciju** su primijenjeni Turnirska selekcija (`tools.selTournament`) za jedno-kriterijsku optimizaciju, te `tools.selNSGA2` za više-kriterijsku optimizaciju. Kao operator **križanja** korišteno je križanje u dvije točke (`tools.cxTwoPoint`), koje razmjenjuje segmente između dva roditeljska kromosoma, dok je za **mutaciju** korištena 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 i obrada rezultata

Za analizu i prikaz rezultata dobivenih optimizacijom korištene su biblioteke `pandas` za tabličnu obradu podataka te `Seaborn` i `Matplotlib` [23, 24] za grafičku vizualizaciju. Ovaj pristup omogućio je sustavnu prezentaciju statistički obrađenih podataka kroz detaljne tablice te vizualnu usporedbu modela pomoću stupčastih

i raspršenih dijagrama. Posebno je značajan prikaz Paretovog fronta, koji jasno ilustrira kompromis (*trade-off*) između maksimizacije ROI-a i minimizacije trajanja, pružajući intuitivan uvid u kvalitetu rješenja dobivenih više-kriterijskom optimizacijom.

Kombinacija alata omogućila je jasnu i preglednu prezentaciju rezultata dobivenih iz eksperimentalnih scenarija.

Ključni vizualni elementi korišteni u ovom radu uključuju:

**Tablične prikaze:** Detaljne tablice s konačnim, statistički obrađenim rezultatima usporedbe različitih optimizacijskih scenarija, uključujući osnovne metrike poput prosječnog ROI-a, prosječnog trajanja te raspona vrijednosti.

**Stupčaste dijagrame:** Koristili su se za vizualnu usporedbu prosječnih vrijednosti (*ROI* i trajanje) između različitih metodologija optimizacije, omogućujući brzu identifikaciju učinkovitijih pristupa.

**Raspršene dijagrame (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. U ovisnosti o eksperimentalnoj seriji, broj jedinstvenih projektnih aktivnosti varirao je od 10 do 100. 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 (BUDGET) bio je skaliran u skladu sa složenošću problema.

### 5.2 Eksperimentalni dizajn

Kako bi se osigurala metodološka ispravnost i izbjegli proizvoljni zaključci, istraživanje je provedeno kroz dvo-fazni 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 reprezentativni tip problema (50 aktivnosti). 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 kao osnova za provođenje konačne usporedbe triju različitih optimizacijskih scenarija i evaluaciju glavnih hipoteza rada na problemima različite skale i restriktivnosti.

### 5.2.1 Istraživačke hipoteze

Na temelju teorijske podloge i postavljenih istraživačkih pitanja iz Uvoda, definirane su sljedeće tri glavne hipoteze koje će se provjeriti kroz eksperimente:

**H1 (Hipoteza o Skalabilnosti):** Povećanjem složenosti problema (broja aktivnosti), performanse modela temeljenog na nasumičnoj pretrazi (Random Search) će se značajno smanjiti u usporedbi s modelima temeljenim na genetskim algoritmima.

**H2 (Hipoteza o Kompromisu):** Hibridni ‘GA+MC’ model će, za razliku od klasičnog ‘GA (samo ROI)’ modela, uspješno identificirati rješenja koja predstavljaju superioran kompromis između profitabilnosti (ROI) i rizika (trajanje projekta), posebno na problemima veće složenosti.

**H3 (Hipoteza o Utjecaju Ograničenja):** Restriktivnost problema, specifično kroz promjenu raspoloživog budžeta, značajno utječe na performanse i stabilnost optimizacijskih modela, pri čemu se očekuje da će ekstremna ograničenja predstavljati najveći izazov za najsloženije modele.

## 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.

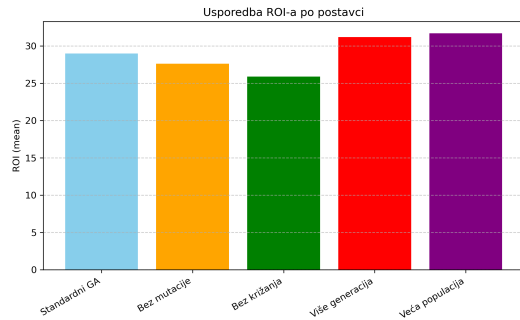
**Metodologija:** Provedena je ablacijska studija s pet različitih konfiguracija, gdje je svaka pokrenuta 10 puta ( $RUNS = 10$ ) radi statističke pouzdanosti. Testirane konfiguracije su bile: *Standardni GA*, *Bez mutacije*, *Bez križanja*, *Više generacija* i *Veća populacija*.

**Rezultati i diskusija:** Rezultati ablacijske studije prikazani su u Tablici 2 te grafički na Slici 3.

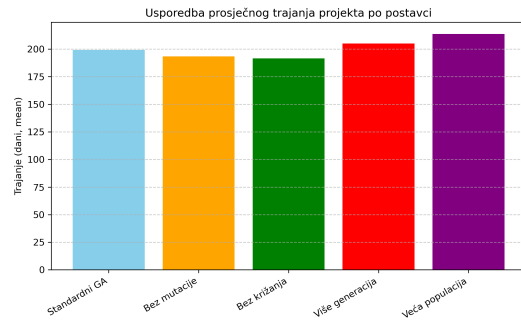
Tablica 2: Rezultati ablacijske studije za parametre GA.

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
<b>Veća populacija</b>	<b>31.683</b>	<b>0.720</b>	<b>213.694</b>	<b>5.574</b>

Analiza rezultata potvrđuje obje početne hipoteze. Uklanjanje križanja drastično smanjuje performanse, potvrđujući da je rekombinacija dobrih rješenja ključan mehanizam pretrage. Povećanje računalnih resursa, posebno kroz veću populaciju, dovodi do superiornih i statistički stabilnijih rješenja. Zanimljivo je primijetiti da konfiguracije s najvišim ROI-em rezultiraju i najdužim prosječnim trajanjem, što stvara prirodni kompromis (*trade-off*) između profita i rizika, koji će biti predmet analize u sljedećem eksperimentu.



(a) Usporedba prosječnog ROI-a.



(b) Usporedba prosječnog trajanja.

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, itd.) poslužili su kao osnova za definiranje parametara u drugoj fazi istraživanja, uz nužne prilagodbe resursa s obzirom na složenost problema.

## 5.4 Eksperiment 2: Usporedna analiza optimizacijskih modela

Druga faza istraživanja čini ključnu eksperimentalnu provjeru glavne hipoteze rada. Koristeći kalibrirane parametre iz Eksperimenta 1, provedena je sustavna usporedba triju razvijenih modela prema planu definiranom u Tablici 3.

Tablica 3: Plan naprednih eksperimenata.

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

### 5.4.1 Rezultati

Svi rezultati dobiveni provođenjem Eksperimenta 2 sažeti su u Tablici 4. Ova tablica predstavlja temelj za daljnju diskusiju.



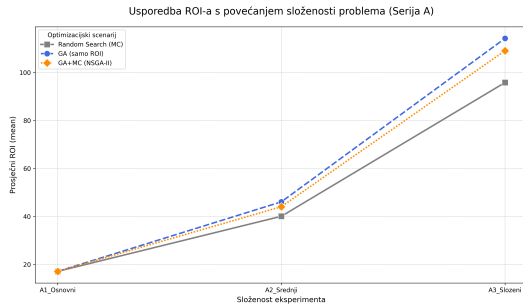
Tablica 4: Konačni rezultati usporedne analize optimizacijskih modela.

Eksperiment	Scenarij	ROI_mean	ROI_std	Trajanje_mean	Trajanje_std
A1_Osnovni	Random Search (MC)	17.140	3.55e-15	144.151	0.911
A1_Osnovni	GA (samo ROI)	17.140	3.55e-15	143.103	1.239
A1_Osnovni	GA+MC (NSGA-II)	17.140	3.55e-15	142.003	0.372
A2_Srednji	Random Search (MC)	40.108	0.703	341.024	9.405
A2_Srednji	GA (samo ROI)	46.125	0.412	363.632	7.843
A2_Srednji	GA+MC (NSGA-II)	44.099	0.980	319.210	13.171
A3_Složeni	Random Search (MC)	95.835	1.468	715.604	10.451
A3_Složeni	GA (samo ROI)	114.224	0.891	792.300	11.933
A3_Složeni	GA+MC (NSGA-II)	109.095	2.008	681.565	21.733
B1_Restriktivan	Random Search (MC)	24.120	1.846	197.621	20.181
B1_Restriktivan	GA (samo ROI)	37.976	0.567	253.562	12.386
B1_Restriktivan	GA+MC (NSGA-II)	17.711	17.728	50104.382	49894.619
B3_Labav	Random Search (MC)	71.379	1.114	536.901	16.247
B3_Labav	GA (samo ROI)	79.065	0.518	562.191	9.345
B3_Labav	GA+MC (NSGA-II)	76.949	0.487	526.612	10.602

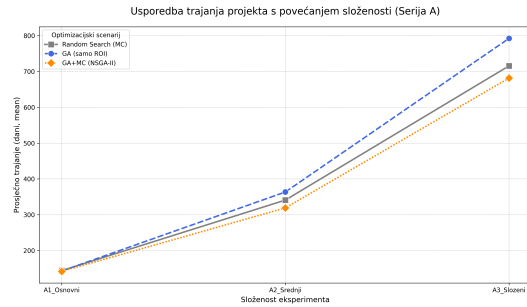
### 5.4.2 Diskusija rezultata

Dobiveni rezultati analizirani su kroz tematske cjeline, s ciljem odgovaranja na postavljena istraživačka pitanja.

**Analiza Skalabilnosti (Serija A)** Kao što je vidljivo na Slici 4, porast složenosti problema drastično utječe na performanse modela. Jaz u **ROI\_mean** vrijednostima eksponencijalno raste u korist genetskih algoritama, potvrđujući hipotezu o nužnosti inteligentne pretrage (H1). Ovakav nalaz, gdje metaheuristički pristupi značajno nadmašuju nasumičnu pretragu na složenim problemima, u skladu je s rezultatima koje su dobili i drugi istraživači u srodnim domenama primjene [13]. Istovremeno, analiza trajanja otkriva postojanje kompromisa: hibridni model GA+MC (NSGA-II) konzistentno identificira rješenja sa značajno nižim prosječnim trajanjem, potvrđujući hipotezu H2.



(a) Usporedba prosječnog ROI-a.

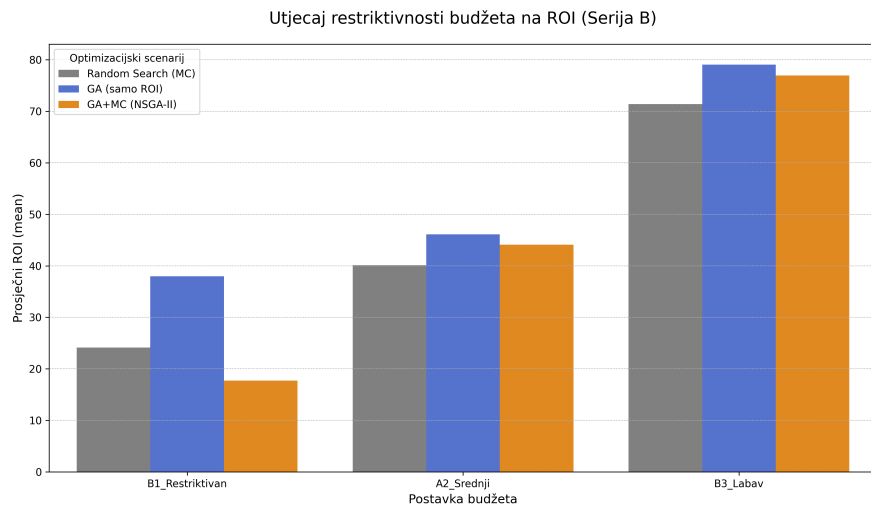


(b) Usporedba prosječnog trajanja.

Slika 4: Grafički prikaz rezultata Serije A: Usporedba modela u uvjetima rastuće složenosti.

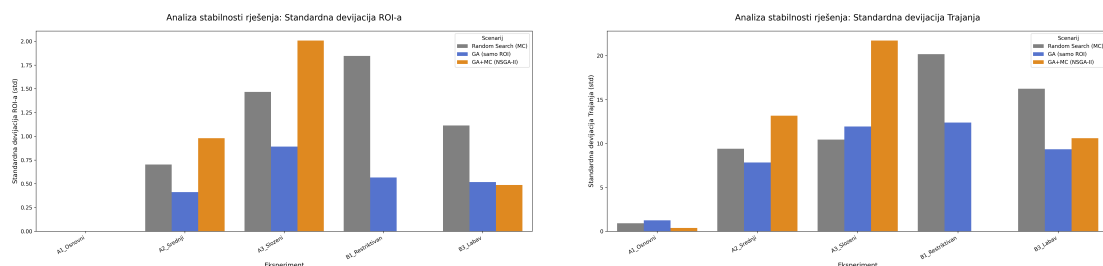
**Analiza Utjecaja Ograničenja (Serija B)** Slika 5 ilustrira ponašanje modela pod različitim proračunskim pritiskom. Najvažniji nalaz dolazi iz eksperimenta s restriktivnim budžetom (B1), gdje GA+MC (NSGA-II) pokazuje iznimnu krhkost, ne uspijevajući pronaći valjano rješenje u 50% pokretanja. S druge strane,

jednostavniji GA (samo ROI) pokazuje se vrlo robusnim. 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, potvrđujući hipotezu H3.



Slika 5: Usporedba prosječnog ROI-a modela pod različitim proračunskim ograničenjima (Serija B).

**Analiza Stabilnosti i Pouzdanosti** Grafikoni na Slici 6 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.

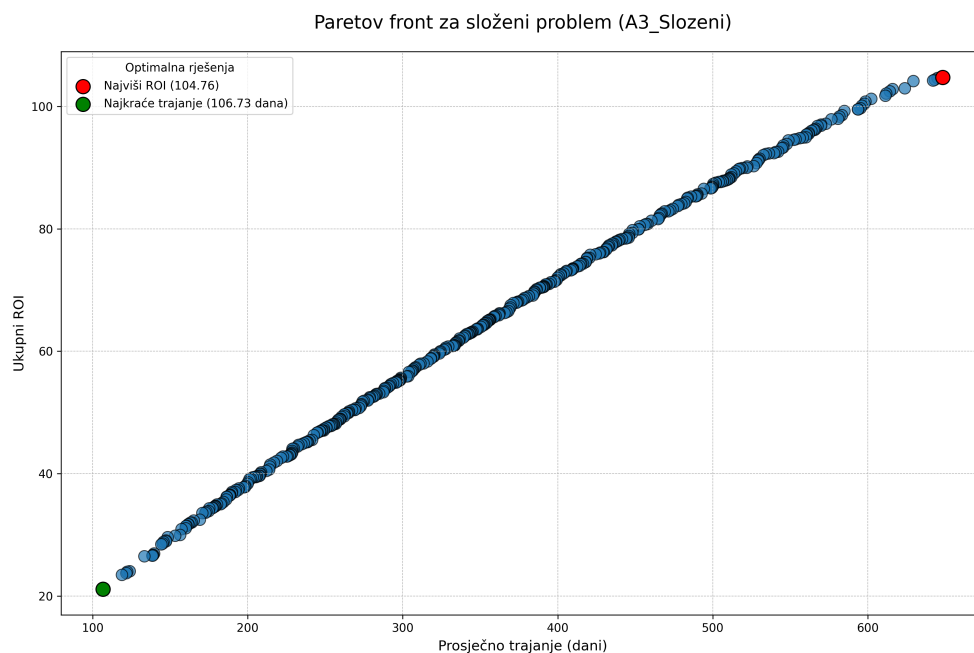


(a) Stabilnost ROI-a.

(b) Stabilnost trajanja.

Slika 6: Grafički prikaz stabilnosti rješenja: Standardna devijacija za ROI i Trajanje.

**Dubinska analiza kompromisa: Pareto front** Raspršeni dijagram na Slici 7 pruža dubinski uvid u srž više-kriterijske optimizacije. On prikazuje Pareto front dobiven iz jednog pokretanja GA+MC (NSGA-II) modela na naj-složenijem problemu (A3). Svaka točka na grafikonu predstavlja jedno optimalno, ne-dominirano rješenje i ilustrira temeljni kompromis (*trade-off*) između profitabilnosti (Y-os) i rizika trajanja (X-os). Pareto front stoga ne nudi jedno "točno" rješenje, već služi kao strateški alat za donošenje odluka.



Slika 7: Pareto front za složeni problem (A3), koji prikazuje kompromis između ROI-a i trajanja.

## 5.5 Sinteza glavnih zaključaka eksperimenata

Provedeni eksperimenti omogućuju donošenje cjelovitih zaključaka o svakom modelu. Random Search (MC) se pokazao korisnim isključivo kao početna točka na jednostavnim problemima, ali je potpuno neadekvatan kao ozbiljan optimizacijski alat za probleme realne veličine. GA (samo ROI) je izuzetno snažan i robustan "profitni maksimizator", idealan u situacijama gdje je financijska dobit jedini kriterij. Konačno, GA+MC (NSGA-II) je sofisticirani "upravitelj rizikom", čija najveća vrijednost leži u pružanju strateških opcija koje balansiraju profit i rizik. Iako je superioran u standardnim i složenim uvjetima, njegova složenost ga čini osjetljivim 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

Ovaj diplomski rad bavio se problemom optimizacije portfelja projektnih aktivnosti u uvjetima nesigurnosti. S ciljem razvoja modela koji donositeljima odluka nudi ne samo profitabilna, već i robusna rješenja, razvijen je i evaluiran hibridni pristup koji integrira snagu genetskih algoritama za pretraživanje složenih prostora rješenja i Monte Carlo simulacije za kvantifikaciju rizika. Kroz sustavni, dvo-fazni eksperimentalni proces, provedena je prvo kalibracija parametara genetskog algoritma, a zatim i detaljna usporedna analiza triju optimizacijskih modela: osnovne metode nasumične pretrage, klasičnog genetskog algoritma usmjerenog isključivo na povrat na investiciju (ROI), te naprednog, više-kriterijskog hibridnog modela (GA+MC).

### 6.1 Glavni nalazi i odgovori na istraživačka pitanja

Provedeni eksperimenti pružili su jasne odgovore na istraživačka pitanja postavljena u uvodu rada.

1. **Superiornost inteligentne pretrage je potvrđena.** Eksperimenti su nedvojbeno pokazali da, iako je na jednostavnim problemima nasumična pretraga mogla pronaći valjana rješenja, njena učinkovitost drastično opada s porastom složenosti. Genetski algoritmi su se pokazali superiornima, pronalazeći rješenja sa značajno višim ROI-em, čime je potvrđena hipoteza H1.
2. **Hibridni model uspješno upravlja kompromisom između profita i rizika.** Hibridni GA+MC model, temeljen na NSGA-II algoritmu, uspješno je identificirao Pareto front optimalnih rješenja. Time je potvrđena hipoteza H2 – model donositelju odluke ne nudi jedno, već čitav spektar strateških opcija koje balansiraju viši ROI s dužim procijenjenim trajanjem, i obrnuto. Vizualizacija Paretoovog fronta pokazala se kao ključan alat za strateško odlučivanje.
3. **Stabilnost i robusnost ovise o kontekstu problema.** Analiza je otkrila ključan, nijansiran nalaz. U standardnim i složenim uvjetima, hibridni GA+MC model nudi rješenja čije je procijenjeno trajanje pouzdanije (niža standardna devijacija). Međutim, pod ekstremnim pritiskom vrlo restriktivnog budžeta, njegova složenost postaje nedostatak, što dovodi do nestabilnosti i neuspjeha u pronalaženju rješenja. U tim uvjetima, jednostavniji, jedno-kriterijski GA pokazao se robusnijim. Time je potvrđena i hipoteza H3.

### 6.2 Doprinos rada

Doprinos ovog rada može se sažeti u tri ključne domene:

- **Metodološki doprinos:** Razvijen je i primijenjen cjelovit, dvo-fazni eksperimentalni okvir za evaluaciju optimizacijskih algoritama, koji uključuje fazu kalibracije (ablacijska studija) i fazu usporedne analize. Pokazano je

da ne postoji univerzalno "najbolji" model, već da izbor ovisi o strateškim prioritetima – maksimizaciji profita ili uravnoteženom upravljanju rizikom.

- **Praktični doprinos:** Rad je demonstrirao kako sinergija genetskih algoritama i Monte Carlo simulacije može pružiti konkretnu vrijednost projektnim menadžerima. Pareto front, kao ključni rezultat hibridnog modela, transformira optimizacijski problem iz potrage za jednim rješenjem u alat za strateško donošenje odluka.

### 6.3 Ograničenja rada

Tijekom istraživanja uočena su i određena ograničenja koja otvaraju prostor za daljnja poboljšanja:

- **Računalna zahtjevnost:** Hibridni GA+MC model, zbog potrebe za izvođenjem stotina Monte Carlo simulacija za svaku evaluaciju jedinke, značajno je računalno skuplji i sporiji od klasičnog GA.
- **Korištenje sintetičkih podataka:** Svi eksperimenti provedeni su na sintetički generiranim podacima. Iako su parametri odabrani da budu realistični, validacija modela na stvarnim projektnim podacima predstavljala bi važan sljedeći korak.
- **Stabilnost pod ograničenjima:** Kao što je pokazao eksperiment B1, napredni više-kriterijski model može biti nestabilan u uvjetima ekstremno restriktivnih resursa.

### 6.4 Preporuke za budući rad

Na temelju provedenog istraživanja i uočenih ograničenja, izdvajaju se sljedeći pravci za budući rad:

- **Hibridizacija s drugim metaheuristikama:** Istraživanje kombinacija s algoritmima rojeva čestica (PSO) ili simuliranim kaljenjem radi potencijalnog poboljšanja brzine konvergencije ili kvalitete rješenja [13].
- **Primjena strojnog učenja:** Korištenje tehnika strojnog učenja za preciznije predviđanje distribucija nesigurnosti iz povijesnih podataka, umjesto oslanjanja na tri točke procjene.
- **Razvoj softverskog alata:** Izrada alata s intuitivnim korisničkim sučeljem koje bi projektnim menadžerima bez ekspertize u optimizaciji omogućilo korištenje ovog modela, s interaktivnom vizualizacijom Paretoovog fronta.

### 6.5 Završna riječ

Ovaj rad je potvrdio da primjena naprednih algoritamskih rješenja nudi značajan potencijal za unapređenje procesa upravljanja projektima u uvjetima nesigurnosti. Učinkovito upravljanje, kako ističe Kerzner [3], u suvremenom okruženju zahtijeva

upravo kombinaciju tradicionalnih i naprednih, kvantitativnih pristupa. Razvijeni i analizirani modeli predstavljaju konkretan doprinos u tom smjeru, pružajući temelj za donošenje odluka koje nisu samo financijski isplative, već i informirane o rizicima koji ih prate.

# 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] Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. PMI, 7th edition, 2021.
- [9] David Hillson. *Managing Risk in Projects*. Routledge, 2009.
- [10] Pete Smith. *Project Uncertainty: Managing Risk in New Ventures*. Routledge, 2014.
- [11] David Vose. *Risk Analysis: A Quantitative Guide*. John Wiley & Sons, 2008.
- [12] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [13] 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.
- [14] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998.
- [15] 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.
- [16] Roger Miller and Donald Lessard. Project risk management using monte carlo simulation. *Project Management Journal*, 40(3):23–33, 2009.

- [17] 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.
- [18] Averill M. Law. *Simulation Modeling and Analysis*. McGraw-Hill Education, 5th edition, 2015.
- [19] Python Software Foundation. Python language reference, version 3.x. <https://www.python.org/>, 2024. Accessed: 2025-07-31.
- [20] 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.
- [21] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020.
- [22] The pandas development team. pandas-dev/pandas: Pandas. feb 2020.
- [23] Michael L. Waskom. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.
- [24] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.



## Popis slika

1	Konceptualni model problema optimizacije portfelja projektnih aktivnosti. . . . .	9
2	Vizualni prikaz toka istraživanja . . . . .	12
3	Grafički prikaz rezultata ablacijske studije za genetski algoritam. . .	19
4	Grafički prikaz rezultata Serije A: Usporedba modela u uvjetima rastuće složenosti. . . . .	20
5	Usporedba prosječnog ROI-a modela pod različitim proračunskim ograničenjima (Serija B). . . . .	21
6	Grafički prikaz stabilnosti rješenja: Standardna devijacija za ROI i Trajanje. . . . .	21
7	Pareto front za složeni problem (A3), koji prikazuje kompromis između ROI-a i trajanja. . . . .	22

## Popis tablica

1	Korištene biblioteke u implementaciji . . . . .	10
2	Rezultati ablacijske studije za parametre GA. . . . .	18
3	Plan naprednih eksperimenata. . . . .	19
4	Konačni rezultati usporedne analize optimizacijskih modela. . . . .	20

## Listings

1	Funkcija za Monte Carlo procjenu trajanja . . . . .	13
2	Više-kriterijska funkcija pogodnosti . . . . .	15