Evolutivni algoritam za upravljanje višeagentskim robotskim sustavom

Ćurković, Petar

Doctoral thesis / Disertacija

2010

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:892168

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb





SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

EVOLUTIVNI ALGORITAM ZA UPRAVLJANJE VIŠEAGENTSKIM ROBOTSKIM SUSTAVOM

DOKTORSKI RAD

PETAR ĆURKOVIĆ

ZAGREB, 2010.

SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

EVOLUTIVNI ALGORITAM ZA UPRAVLJANJE VIŠEAGENTSKIM ROBOTSKIM SUSTAVOM

DOKTORSKI RAD



Mentor:

dr. sc. Bojan Jerbić, red. prof.

Petar Ćurković

Zagreb, 2010.

Podaci za bibliografsku karticu:

UDK: 004.89:681.51

Ključne riječi: evolucijski algoritam, planiranje kretanja, robotika.

Znanstveno područje: Tehničke znanosti.

Znanstveno polje: Strojarstvo.

Institucija u kojoj je rad izrađen: Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu.

Mentor rada: dr. sc. Bojan Jerbić, red. prof.

Broj stranica: 144.

Broj slika: 88.

Broj tablica: 5.

Broj korištenih bibliografskih jedinica: 80.

Datum obrane: 19. studenog 2010.

Povjerenstvo: dr. sc. Josip Kasać, doc. – predsjednik povjerenstva,
dr. sc. Bojan Jerbić, red. prof. – voditelj,
dr. sc. Danica Kragić, red. prof. KTH Stockholm – član povjerenstva.

Institucija u kojoj je rad pohranjen: Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu.

Zahvala

Želio bih najprije zahvaliti mentoru, prof. dr. sc. Bojanu Jerbiću što mi je omogućio dolazak na Fakultet, uveo me u fascinantno istraživačko područje i strpljivo me vodio pri izradi ovoga rada. U tome periodu od gotovo šest godina naučio sam mnogo, na čemu sam posebno zahvalan.

Moj dragi prijatelj, doc. dr. sc. Josip Stepanić pomagao mi je savjetima i mnogim diskusijama da otklonim dvojbe koje su se pri izradi ovoga rada pojavile i time olakšao privođenje ove zahtjevne zadaće kraju.

Prof. dr. sc. Božo Vranješ svojom toplinom i ljudskošću olakšao mi je mnoge trenutke neizvjesnosti, u i izvan poslovnoga okružja i time me potaknuo da u svojim nastojanjima s dodatnim elanom ustrajem.

Dragi suradnici i prijatelji s Katedre, Denis Bašić i Tomislav Stipančić, dipl. ing. nesebično su mi pomagali kada je bilo potrebno, preuzimali moje zadaće i svojim veselim osobnostima bili nepresušan izvor dobrog raspoloženja.

Ministarstvo znanosti, obrazovanja i športa RH, tvrtke Elektrokontakt d.d. te Končar grupa podupirali su istraživanja u okviru kojih je ova disertacija nastala i time pridonijeli njezinoj kvaliteti te im se ovom prigodom zahvaljujem.

Članovi komisije, prof. dr. sc Danica Kragić, *KTH* Stockholm, gdje je tijekom moga boravka dio rada i nastao, i doc. dr. sc. Josip Kasać pažljivim čitanjem teksta i korisnim kritikama doprinijeli su konačnoj kvaliteti ovog rada.

Svojoj obitelji želim zahvaliti za dugogodišnju podršku i odricanje čime su omogućili ostvarenje mnogih mojih nastojanja.

Na kraju, želim zahvaliti svojoj supruzi Nataliji koja je uz mene bila u ovoj osjetljivoj životnoj fazi koja je od nje zahtijevala stalna odricanja te što mi nikada nije dala da to primijetim.

Zagreb, rujna 2010. Petar Ćurković

Za Ivku i Natalију

Man's mind, once stretched by a new idea, never regains its original dimensions.

Oliver Wendell Holmes

SADRŽAJ

Popis slika ... **ix**

Popis tablica ... **xii**

Popis važnijih oznaka i kratica ... **xiii**

Sažetak ... **xvi**

Summary ... **xvii**

1 Uvod ... **1**

1.1 Pregled dosadašnjih spoznaja ... *5*

1.2 Tradicionalne metode planiranja kretanja ... *7*

1.3 Alternativni pristupi planiranju kretanja ... *8*

1.4 Više robota u radnom prostoru ... *10*

1.5 Alternativne metode planiranja putanje u višerobotskim sustavima ... *12*

1.6 Cilj i hipoteza rada ... *14*

1.7 Metodologija i plan istraživanja ... *15*

1.8 Očekivani znanstveni doprinos ... *18*

1.9 Struktura disertacije ... *18*

2 Evolucijski algoritmi ... **21**

2.1 Osnovni evolucijski algoritam ... *24*

2.2 Metode kodiranja kod evolucijskih algoritama ... *25*

2.3 Genetski algoritmi ... *27*

2.3.1 Jednostavan genetski algoritam za planiranje putanje mobilnog robota ... 30

2.3.2 Teoretske osnove genetskih algoritama ... 32

2.3.3 Shema teorem za standardni genetski algoritam ... 33

2.4 Koevolucijski algoritmi ... *35*
2.4.1 Definicija koevolucije ... 35
2.4.2 Kooperativna i kompetitivna koevolucija ... 36
2.4.3 Neke prednosti koevolucije ... 39
2.4.4 Patologije koevolucijskih algoritama ... 40
2.5 Računska složenost evolucijskih algoritama ... *42*
2.6 Evolucijska robotika ... *43*
2.7 Evolucijski algoritmi i višekriterijski optimizacijski problemi ... *48*
3 Razvoj višeagentskog modela upravljanja dvoručnim robotom ... 51
3.1 Jedan agent u radnom prostoru ... *53*
3.1.1 Minimizacija prijeđenog puta ... 57
3.1.2 Minimizacija kuta rotacije u zglobovima ... 61
3.1.3 Višekriterijska funkcija dobrote, minimizacija puta i minimizacija kuta ... 64
3.1.4 Višekriterijska funkcija dobrote; minimizacija puta, minimizacija kuta i minimizacija neravnomjernosti profila brzine ... 66
3.1.5 Slučaj jednog agenta uz prisutnu statičku prepreku ... 70
3.2 Dva agenta u radnom prostoru ... *78*
3.2.1 Agent, višeagentski sustav ... 78
3.2.2 Minimizacije duljine trajektorije ... 81
3.2.3 Minimizacija kuta rotacije u zglobovima ... 83
3.2.4 Višekriterijska funkcija dobrote, minimizacija puta i minimizacija kuta ... 85
3.2.5 Višekriterijska funkcija dobrote; minimizacija puta, minimizacija kuta i minimizacija neravnomjernosti profila brzine ... 86
3.2.6 Puni model – dva agenta u radnom prostoru uz početne i konačne uvjete koji rezultiraju kolizijom tijekom gibanja ... 89
3.2.7 Formalna analiza modela dvoručnog robota ... 90
3.2.8 Rezultati simulacija za dva agenta uz različite početne i konačne uvjete ... 101
3.3 Prilagodba parametara koevolucijskog algoritma ... *109*
4 Validacija i verifikacija predložene metodologije ... 117

4.1 Dvoručni robotski sustav Laboratorija za projektiranje izradbenih i montažnih sustava 117

4.2 Simulacija u fizičkom simulacijskom okružju Roboguide Fanuc Roboticsa 118

5 Zaključak 127

6 Popis literature 133

POPIS SLIKA

Slika 1-1. Jedan dvoručni robotski sustav s pripadajućim radnim prostorima svakog robota. Područje potencijalne kolizije nalazi se u prostoru preklapanja dviju sfera. 3
Slika 1-2. Dva robota koji dijele radni prostor. Laboratorij za projektiranje izradbenih i monatžnih sustava. 4
Slika 1-3. Primjer početne (lijevo) i konačne (desno) konfiguracije za koje decentralizirani pristup ne daje zadovoljavajuće rješenje. 11
Slika 1-4. Dijagram toka istraživanja. 17
Slika 2-1. Primjer konvencionalnog rešetkastog nosača (lijevo) i konstrukcije dobivene genetskim algoritmom (desno). 22
Slika 2-2. Dijagram toka općenitog evolucijskog algoritma. 24
Slika 2-3. Pseudo kôd osnovnog evolucijskog algoritma. 25
Slika 2-4. Kanonski evolucijski algoritmi. 25
Slika 2-5. Primjer kromosoma kodiranog binarnim vrijednostima (kromosom A) i kodiranje realnim vrijednostima (kromosom B). 25
Slika 2-6. Primjeri metoda inicijalizacije početne populacije rješenja. Kvadrati označavaju dvodimenzionalne prostore pretrage. Sivi krugovi predstavljaju fenotip pripadajućeg potencijalnog rješenja. 26
Slika 2-7. Križanje u jednoj točki između dva kromosoma kodirana binarnim nizovima. 29
Slika 2-8. Primjer povezanog (lijevo) i nepovezanog prostora pretrage (desno). 30
Slika 2-9. Mutacija binarnog niza. 30
Slika 2-10. Rješavanje problema planiranja putanje jednostavnog agenta primjenom genetskog algoritma. 31
Slika 2-11. Putanje dobivene genetskim algoritmom. 32
Slika 2-12. Dijagram toka osnovnog koevolucijskog algoritma. 36
Slika 2-13. Vrednovanje pojedinca kod standardnog evolucijskog (lijevo) i koevolucijskog algoritma (desno). 38
Slika 2-14. Prikaz općenitog višekriterijskog optimizacijskog problema. 49
Slika 3-1. Jedan agent u radnom prostoru uz prisutnu prepreku. 53
Slika 3-2. Rješenje uz kodiranje kromosomom duljine $l = 2$. 54
Slika 3-3. Analiza kretanja referentne točke robota. 55

Slika 3-4. Rekonstrukcija konfiguracija s po volji finom diskretizacijom *T*. Iz diskretnog izlaza EA, odabranog tako da se konvergencija ostvaruje u polinomijskom vremenu, interpolacijom odgovarajućim polinomom dolazi se do kontinuirane raspodjele kuta svakog zgloba ovisno o vremenu. Iz ove je raspodjele moguće rekonstruirati konfiguracije u proizvoljnom trenutku *T*. 56
Slika 3-5. Rješenja pronađena EA u slučaju minimizacije prijeđenog puta. 59
Slika 3-6. Konvergencija EA u slučaju minimizacije prijeđenog puta. 60
Slika 3-7. Nastavak prikaza konvergencije EA u slučaju minimizacije prijeđenog puta. 60
Slika 3-8. Ovisnost kuta zakreta o vremenu za dva slučaja konvergencije..... 61
Slika 3-9. Rješenja pronađena EA u slučaju minimizacije kuta. 62
Slika 3-10. Konvergencija EA u slučaju minimizacije kuta. 63
Slika 3-11. Promjena kuta zakreta u zglobovima u vremenu..... 63
Slika 3-12. Rješenje EA uz složenu optimizaciju prijeđenog puta i ukupnog kuta rotacije, slučaj I. 64
Slika 3-13 Rješenje EA uz složenu optimizaciju prijeđenog puta i ukupnog kuta rotacije, kriterij puta nije optimalno zadovoljen, slučaj II. 65
Slika 3-14. Ovisnost kutova zakreta o vremenu za slučaj I. 65
Slika 3-15. ovisnost kutova zakreta o vremenu za slučaj II. 66
Slika 3-16. Konvergencija EA, slučaj I. 66
Slika 3-17. Konvergencija EA, slučaj II. 67
Slika 3-18. Rješenje EA uz složenu funkciju dobrote, kriteriji prijeđenog puta, kuta zakreta i profila brzine. 68
Slika 3-19. Konvergencija EA za slučaj s tri minimizacijska kriterija. 69
Slika 3-20. Promjena kuta zakreta zglobova 1 i 2 uz minimizaciju prijeđenog puta, minimizaciju kuta zakreta i minimizaciju nejednolikosti brzine. 70
Slika 3-21. Rješavanje problema planiranja putanje uz jednu prisutnu prepreku, duljina gena *l*=8. 73
Slika 3-22. Rješavanje problema planiranja putanje uz jednu prisutnu prepreku, duljina gena *l*=6. 73
Slika 3-23. Suboptimalno rješenje na izlazu evolucijskog algoritma. 74
Slika 3-24. Kontinuirana promjena kuta zakreta zgloba *i* u vremenu za niz konfiguracijama prikazan slikom [Slika 3-21]..... 74
Slika 3-25. Kretanje robota rekonstruirano s vremenskim korakom $T = 0.01$ s. 75
Slika 3-26. Kretanje robota rekonstruirano s vremenskim korakom $T = 0.1$ s. 76

Slika 3-27. Konvergencija EA za rješenje prikazano slikom [Slika 3-22]..... 76
Slika 3-28. Struktura općenitog višeagentskog sustava. 78
Slika 3-29. Agenti s pripadajućom funkcijom dobrote u uvjetima promjenjive okoline. 79
Slika 3-30. Dva agenta u radnom prostoru. 80
Slika 3-31. Evolucija dva robota u radnom prostoru. Lijevi robot, R_1 ima ugrađen elitizam i dostiže globalno optimalnu vrijednost. 81
Slika 3-32. Konvergencija dviju populacija uz prisutnost / izostanak elitizma. 81
Slika 3-33. Evolucija dva robota u radnom prostoru, oba robota imaju ugrađen elitizam. 82
Slika 3-34. Konvergencija dviju populacija uz prisutnost elitizma kod obje populacije. 83
Slika 3-35. Evolucija dva robota u radnom prostoru, kriterij kuta rotacije u zglobovima. 84
Slika 3-36. Konvergencija populacija prema kriteriju puta. 84
Slika 3-37. Evolucija dva robota u radnom prostoru prema kriterijima minimizacije puta i kuta rotacije 85
Slika 3-38. Konvergencija populacija prema kriterijima puta i kuta rotacije. 85
Slika 3-39. Kod robota R_1 implementirana su tri kriterija, put, kut rotacije i nejednolikost brzine, dok su kod robota R_2, desno, implementirana samo dva kriterija, put i kut. 86
Slika 3-40. Konvergencija populacije za odabrane kriterije. 86
Slika 3-41. Dva robota, tri optimizacijska kriterija, put, kut rotacije i nejednolikost brzine. 87
Slika 3-42 Prikaz konvergencije za odabrane kriterije 87
Slika 3-43. Zakret zgloba u ovisnosti o vremenu 89
Slika 3-44. Kretanje robota sa smanjenim vremenom uzorkovanja, $T = 0.01$ s. 89
Slika 3-45. Definicija dva robota u radnom prostoru. 91
Slika 3-46. Presjecište dva bazna segmenta. 92
Slika 3-47. Slučaj presjecišta vršnog segmenta robota R_1 i baznog segmenta robota R_2. 92
Slika 3-48. Slučaj presjecišta vršnog segmenta robota R_2 i baznog segmenta robota R_1. 93
Slika 3-49. Presjecište dva vršna segmenta. 94
Slika 3-50. Površine robota bez dodirnih točaka 96
Slika 3-51. Jedno presjecište. 96
Slika 3-52. Opći slučaj, potrebno je konstruirati putanju. 97
Slika 3-53. Definicija težišta spojnice i vektora položaja. 98
Slika 3-54. Rotacija spojnice s oko težišta T. 100
Slika 3-55. Dva SCARA robota u radnom prostoru. 101
Slika 3-56. Slučaj s dva agenta, vrednovanje kolizije nije uključeno u izračun dobrote. 101
Slika 3-57. Slučaj s dva agenta, vrednovanje kolizija uključeno na agentu R_1. 102

Slika 3-58. Uspješno rješenje planiranja putanje za dva agenta u radnom prostoru. 103
Slika 3-59. Prikaz konvergencije najboljih pojedinaca iz dviju populacija. 103
Slika 3-60. Kontinuirano kretanje uz prilagođavanje kretanja radi izbjegavanja kolizija. 104
Slika 3-61. Prikaz promjene zakreta kuta u zglobovima robota R_1 i robota R_2..... 104
Slika 3-62. Prikaz nedominiranih rješenja za agenta R_1 u radnom prostoru tijekom evolucije; a) $G = 10$ generacija; b) $G = 100$ generacija i c) $G = 400$ generacija – rješenje prikazano slikom [Slika 3-60]. Crveno – nedominirana rješenja; plavo – dominirana rješenja. Skale vrijednosti su normirane u području [0,1]. Veličina populacije $N = 60$; duljina kromosoma $l = 8$. 106
Slika 3-63. Prostorni prikaz konvergencije. Slika a) $G = 1$; b) $G = 50$; c) $G = 200$ i d) $G = 400$. Lijeva strana slike pokazuje kombinaciju kriterija nejednolikosti, udaljenosti i kuta rotacije, desna strana slike pokazuje kombinaciju kriterija nejednolikosti, udaljenosti i broja kolizija. 108
Slika 3-64. Identifikacija povoljne vjerojatnosti mutacije. 113
Slika 3-65. Konvergencija i razina pouzdanosti za različite metode definiranja parametara koevolucijskog algoritma. 114
Slika 4-1. Roboti korišteni pri provedbi inicijalnih eksperimenata. 117
Slika 4-2. Princip rada fanucovog *Roboguide* simulacijskog okružja. 119
Slika 4-3. Početna (lijevo) i konačna (desno) konfiguracija robota 120
Slika 4-4. Rješenje problema planiranja putanje za desnog robota. A-početna konfiguracija, F-konačna konfiguracija. 121
Slika 4-5. Rješenje problema planiranja putanje za oba robota. A-početna konfiguracija, F-konačna konfiguracija. 122
Slika 4-6. Validacija predložene metodologije. 125

POPIS TABLICA

Tablica 2-1. Učinak predloženog genetskog algoritma uz različite parametre. 32
Tablica 3-1. Vrijednosti parametara EA za slučaj minimizacije puta. 58
Tablica 3-2. Izabiranje parametara koevolucijskog algoritma. 111
Tablica 4-1. Parametri početne i konačne konfiguracije lijevog robota..... 123
Tablica 4-2. Parametri početne i konačne konfiguracije desnog robota 124

POPIS VAŽNIJIH OZNAKA I KRATICA

Oznaka	Značenje	Mjerna jedinica
	Latinične oznake	
a	Ubrzanje	m/s^2
$\vec{a}$	Bazni vektor segmenta robota R_1	-
$\vec{b}$	Vršni vektor segmenta robota R_1	-
$\vec{c}$	Bazni vektor segmenta robota R_2	-
C	Cityblock udaljenost	m
C_0	Linearna klasa interploacijske funkcije	-
$\mathcal{C}$	Skup svih konfiguracija robota	-
$\mathcal{C}_f$	Slobodan konfiguracijski prostor	-
$\vec{d}$	Vršni vektor segmenta robota R_2	-
$\mathcal{F}$	Fenotipski prostor	-
$\tilde{f}$	Prosječna dobrota populacije	-
G	Trenutna generacija	-
$\mathcal{G}$	Genotipski prostor	-
H	Shema	-
J_i	Stupanj slobode i	-
l	Duljina kromosoma	-
m	Diskretno vrijeme	-

N	Veličina populacije	-
$\mathbb{N}$	Skup prirodnih brojeva	-
p_c	Vjerojatnost križanja	-
P_d	Vjerojatnost uništavanja sheme	-
p_m	Vjerojatnost mutacije	-
P_s	Vjerojatnost stvaranja sheme	-
$\mathcal{P}^*$	Pareto optimalan skup	-
$\mathcal{PF}^*$	Pareto fronta	-
$\mathcal{PR}^*$	Skup presjecišta	-
$\mathbb{R}$	Skup realnih brojeva	-
$\mathcal{R}$	Mapa putova	-
T	Vrijeme između dvije uzastopne konfiguracije	s
v	Brzina	m/s
V_{max}	Maksimalna brzina gibanja referentne točke	m/s
w_1	Težinski faktor 1	-
w_2	Težinski faktor 2	-

Grčke oznake

β	Trenutni segment robota	-
ΔS	Razlika prijeđenog puta	m
$\Delta\varphi_x$	Promjena kuta x	rad
Θ	Funkcija potrebnog broja računskih operacija	-
θ	Prikloni kut spojnice	rad
Λ	Prostor rješenja	-

μ	Očekivanje	-
ξ	Bezdimenzijska značajka konfiguracije	-
Π	Spojnica referentnih točaka	-
σ	Standardna devijacija	-
Φ	Funkcija dobrote	-
φ_1	Kut između baznog segmenta R_1 i apscise	rad
φ_2	Kut između vršnog segmenta R_1i apscise	rad
φ_3	Kut između baznog segmenta R_2 i apscise	rad
φ_4	Kut između vršnog segmenta R_2 i apscise	rad
Ω	Alfabet kodiranja	-

Popis kratica

$2R$	Robotska konfiguracija sa dva rotacijska zgloba
EC	*Evolutionary Computation* (evolucijsko računarstvo)
EP	Evolucijsko programiranje
ES	Evolucijske strategije
GA	Genetski algoritam
GP	Genetsko programiranje
MOEA	*Multi Objective Evolutionary Algorithm* (višekriterijski evolucijski algoritmi)
NP	Nedeterminističko polinomijsko vrijeme
PRM	*Probabilistic Roadmap Method* (stohastičkametoda putova)
SCARA	*Selective Compliant Assembly Robot Arm* (karakteristična robotska konfiguracija)

SSG	Broj stupnjeva slobode gibanja
VOP	Višekriterijski optimizacijski problem

Sažetak

U ovoj disertaciji razvijena je metodologija za planiranje kretanja dva robota 2*R* konfiguracije koji dijele radni prostor i predstavljaju dinamičku prepreku jedan drugome. Ovaj problem spada u klasu *NP* kompletnih problema, čije je rješavanje egzaktnim matematičkim algoritmima s poznatim gornjim granicama vremenske složenosti ograničeno primjenjivo.

Razvijena metodologija planiranja kretanja temeljena je na istodobnom rješavanju problema planiranja kretanja za dva robota paralelnim, dijelom neovisnim evolucijskim – koevolucijskim algoritmom. Postavljena je i potvrđena hipoteza o mogućnosti vremenski efikasnog planiranja kretanja za dva robota temeljena na predloženom koevolucijskom algoritmu.

S ciljem potvrđivanja hipoteze, razvijeno je simulacijsko okružje u kojemu je, za različite početne i konačne konfiguracije, moguće planirati kretanje za jednog, ili više robotskih agenata u radnom prostoru koji može uključivati stacionarne prepreke. Određen je skup kriterija kojima je moguće vrednovati ponašanje agenata, te su utvrđene prirode veza među pojedinim kriterijima. Skup kriterija čine kriteriji broja sudara, ukupno prijeđenog puta, kuta zakreta u zglobovima te neravnomjernosti profila brzine. Izvršena je formalna analiza sustava sastavljenog od dva robota 2*R* konfiguracije te su izvedeni analitički uvjeti za postojanje sudara među robotima. Izvedeni su izrazi za izračun potrebne duljine gena evolucijskog algoritma.

Na temelju ovih spoznaja formulirana je funkcija dobrote i odgovarajuća metoda izbora agenata, čime je omogućena njihova paralelna evolucija i međusobna prilagodba, koja rezultira neprekinutim, vremenski i energetski učinkovitim gibanjem bez sudara od početne do konačne konfiguracije u radnom prostoru.

Sa svrhom dodatne uštede procesorskog vremena, predložena je metoda diskretizacije kretanja robota te komplementarna metoda rekonstrukcije kretanja s proizvoljno malim vremenskim intervalom između dvije uzastopne robotske konfiguracije.

Metodologija je verificirana na realnom robotskom sustavu, sastavljenom od dva robota za koja je problem planiranja kretanja razvijenom metodologijom uspješno riješen.

Ključne riječi: evolucijski algoritam, planiranje kretanja, robotika.

Summary

In this thesis a method is developed to solve the problem of path planning for two 2*R* type robots, which share workspace and thus present dynamic obstacle to each other. This problem, according to its computational complexity, belongs to *NP* complete class problems. Solving such problems with exact mathematical algorithms with known upper time complexity bound is limitedly applicable.

The motion planning methodology developed in this thesis is based on concurrent path planning of both robots using parallel, partly independent evolutionary – coevolutionary algorithms. Time efficient planning is possible for two robots using developed algorithms what proves the hypothesis formulated in the dissertation.

To prove the hypothesis, a simulation environment is developed in the first step, in which path planning is possible for arbitrary setup of initial and final configurations of the robots. It is possible to conduct path planning either for one robotic agent or for more, including presence of arbitrarily distributed obstacles. Set of criteria is determined to evaluate the evolution of agents, and the nature of connections between different conflicting criteria is revealed. Number of collisions, total path, joint rotation angle and velocity profile of the end effector are the criteria to be optimized. Formal analysis of the system composed of two 2*R* type robots is conducted and analytical conditions for collision occurrence are derived.

Based on these understandings, the fitness function is formulated with appropriate selection mechanisms, what enabled parallel evolution and adaptation between the two robotic agents. The result is continuous, time and energy efficient motion without collisions for the two agents in the given workspace.

To additionally optimize processing time; a motion discretization method is proposed together with complementary motion reconstruction method to execute the motion with arbitrarily small time step between two consecutive configurations.

The methodology proposed in this thesis is verified on a real robotic system composed of two robots for which the path planning problem is successfully solved.

Key words: evolutionary algorithm, motion planning, robotics.

1 Uvod

Procjene pokazuju kako se u bliskoj budućnosti očekuje izražen rast tržišta uslužne robotike [1]. Unatoč značajnim resursima uloženim u razvoj robota prihvatljivih širokom krugu korisnika, ovi roboti i dalje nisu zastupljeni s udjelom koji je očekivan na temelju provedenih različitih recentnih analiza. Izazov robotike danas je razvoj robota sposobnih za opsežnu interakciju s ljudima. Pri tome se ističu dva problema: razvoj dovoljno robusnih upravljačkih algoritama, te razvoj komunikacijskog sučelja prihvatljivog najširem krugu korisnika.

Okružje u kojem trebaju djelovati roboti koji izlaze iz okvira proizvodnih okolina, promjenjivo je i nekarakterizirano, dakle kompleksno i neprilagođeno robotu. Robot koji treba izvršiti zadaću u takvoj okolini, pored znanja o samom zadatku mora imati sposobnost spoznaje, interpretacije i interakcije s okolinom. Razvojem kompleksne senzorike, u prvom redu eksteroceptičnih senzora poput strojnog vida, osigurane su pretpostavke za akviziciju informacija iz okoline. Razvojem proprioceptičnih senzora poput *GPS* senzora, osigurava se bolja lokalizacija robota u prostoru. Ipak, ostaju prisutni problemi brzine obrade informacija potrebnih za planiranje budućih akcija robota te problemi pojave šuma pri akviziciji podataka.

Pokazalo se da razvoj općenitog monolitnog upravljačkog programa za robote koji djeluju u opisanim uvjetima nije moguć zbog nesagledivog povećanja kompleksnosti upravljačkog algoritma [2]. Naime, nemoguće je strukturama *ako-onda* obuhvatiti sve situacije u kojima se robot tijekom djelovanja može naći. Iako je takav pristup prihvatljiv u industrijskim okolinama, gdje je prostor stanja robota ograničen na skup unaprijed poznatih stanja, koje se rješavaju skupom poznatih djelovanja, roboti najnovije generacije moraju počivati na drugačijim upravljačkim paradigmama.

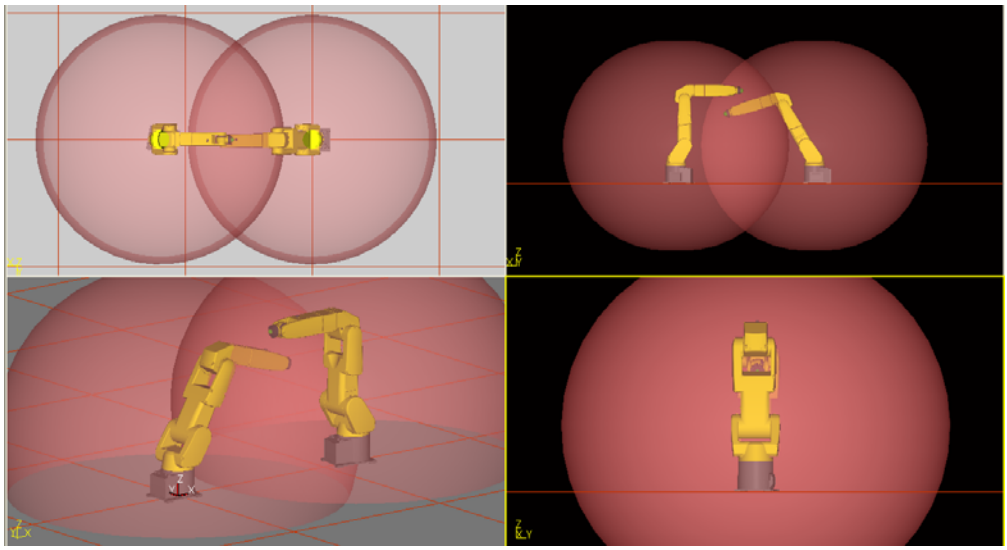
Područje istraživanja inteligentnih robota vrlo je aktivno, interdisciplinarno i obuhvaća, među ostalim: razvoj sklopovlja – senzoriku, upravljačka računala, materijale, i upravljačku stranu: arhitekture, algoritme, protokole. Kako se očekivanja postavljena na razvijene robote sve više približavaju željenim svojstvima živih organizama, u razvoju takvih robota sudjeluju i istraživači iz područja biologije, psihologije, neuroznanosti.

Uspoređujući prirodne, naizgled jednostavne sustave poput mravljih zajednica, iznenađuje koliko su sposobnosti robota koji se danas nalaze na najvišem stupnju razvoja ograničene. Radnje poput kretanja po nekarakteriziranoj podlozi, izbjegavanja prepreka, praćenja pomičnog cilja, prepoznavanja, zajedničkog nošenja tereta, snalaženja i lokalizacije u nepoznatoj okolini samo su neki od problema s kojima se istraživači danas intenzivno bave.

Pretpostavka za uspješno rješavanje zadatka pri radu autonomnog robota je sposobnost samostalnog kretanja u nepoznatoj okolini. Planiranje putanje općenito je težak problem, a posebice za sustave koji sadrže veći broj robota koji dijele radni prostor. Rješavanje problema planiranja putanje u takvoj okolini je *NP*-kompletan problem[i] čija kompleksnost raste eksponencijalno s brojem stupnjeva slobode gibanja [3], [4], [5]. Planiranje putanje u industrijskim je okolinama bitno zbog težnje zamjene ljudskog rada robotskim, posebice kod monotonih ili opasnih radnih mjesta. Unatoč ovoj težnji, ljudi i roboti najčešće dijele radni prostor, pri čemu su roboti unaprijed programirani i njihovo je kretanje moguće predvidjeti. No, u slučaju implementacije autonomnih robota, kretanje robota nije unaprijed poznato, a koliziju s čovjekom, koja najčešće rezultira ozljedom potrebno je bezuvjetno izbjegavati.

Većinom uzroci poteškoćama planiranju putanje leže u nekarakteriziranoj okolini. Ukoliko robot koristi prethodno učitanu mapu prostora, izgledno je da je mapa nepotpuna, ne sadrži informacije o promjenama prostornih odnosa koje mogu nastupiti među objektima, također i sama metrika mape može se razlikovati od stvarnog stanja. Sljedeći problem nalazi se u činjenici da su informacije ekstrahirane iz okoline osjetilima robota ograničeno pouzdane.

[i] Pojam iz računske teorije složenosti, odnosi se na klasu problema koji su rješivi u polinomijskom vremenu nedeterminističkim Turingovim strojem.



Slika 1-1. Jedan dvoručni robotski sustav s pripadajućim radnim prostorima svakog robota. Područje potencijalne kolizije nalazi se u prostoru preklapanja dviju sfera.

Ograničen prostor očitanja u kombinaciji s varijacijama koje je nemoguće unaprijed modelirati, poput promjene intenziteta svjetla, temperature, pojave vibracija neminovno dovodi do šuma kod očitanja informacija iz okoline. Treće, stvarne okoline karakterizirane su kompleksnom i nepredvidivom dinamičkom prirodom: objekti se mogu pomicati, okolina se može mijenjati čak i u naizgled stabilnim detaljima primjerice uslijed promjene vremenskih uvjeta ili godišnjeg doba. I četvrti problem, samo izvođenje zadaća izvršnim elementima nije nije potpuno pouzdano, kotač može proklizati, hvataljka može izgubiti kontakt s objektom.

Predmet istraživanja ove disertacije je razvoj algoritma za upravljanje dvoručnim robotskim sustavom [Slika 1-1]. Dvoručna robotska konfiguracija inspirirana je anatomijom prirodnih sustava koji su većinom simetrični pri čemu se udovi pojavljuju u parovima. Primjenom dvije ruke, čovjek je sposoban izvoditi zadatke većeg stupnja složenosti u odnosu na zadatke koje bi mogao rješavati jednom rukom. Promatrano sa stanovišta robotskih sustava, dvoručni robotski sustav, uz određena geometrijska ograničenja svodi se na višerobotski sustav. Pri tome jedan robot predstavlja pokretnu, dinamičku prepreku drugom robotu s kojim dijeli radni prostor.

Ovaj scenarij pojavljuje se kod problema uzimanja po slučaju raspoređenih objekata iz prostora kojemu pristup imaju izvršni članovi dva robota, eng. *bin picking problem*. Ovaj se problem uspješno rješava primjenom jednog robota i stereovizijskog sustava u općem slučaju. U slučaju kada dva ili više robota imaju pristup objektima, problem se komplicira te je

potrebno uvesti prioritizaciju djelovanja robota (u kojem se slučaju smanjuje vremenska učinkovitost) ili rješavati istodobno za dva robota problem planiranja putanje. Postav kod kojega se ovaj problem želi integralno riješiti, a za čije je rješenje neophodna komponenta istodobnog planiranja putanje za dva robota nalazi se u Laboratoriju za projektiranje izradbenih i montažnih sustava Fakulteta strojarstva i brodogradnje u Zagrebu.



Slika 1-2. Dva robota koji dijele radni prostor. Laboratorij za projektiranje izradbenih i monatžnih sustava.

Zadatak je pronaći takve putanje za svaki robot, koje u prvom redu eliminiraju pojavu kolizije među segmentima robota, ali istovremeno dovoljno dobro zadovoljavaju ostale kriterije, poput ravnomjernosti profila brzine i ubrzanja i minimizacije utrošene energije. Prostor stanja dva robota dvoručne robotske konfiguracije vezan je i suštinski se razlikuje od slučaja kada se u radnom prostoru robota nalazi pokretni objekt na koji nije moguće utjecati.

Stoga se predlaže rješavanje problema planiranja putanje ovog inherentno distribuiranog sustava primjenom koevolucijskog algoritma. Predmet optimizacije dvama paralelnim, dijelom neovisnim evolucijskim procesima su diskretni nizovi konfiguracija lijevog i desnog robota, koji određuju tranziciju svakog robota iz poznatih početnih u poznate konačne konfiguracije. Predloženim se pristupom uzima u obzir i koristi činjenicom da su dva robota međusobno neovisni i upravljani odvojenim upravljačkim računalima. Time se štedi vrijeme potrebno za izvođenje računskih operacija i povećava efikasnost sustava kao cjeline.

1.1 Pregled dosadašnjih spoznaja

Iako jedinstvena općeprihvaćena definicija robota ne postoji, svim definicijama zajedničko je da na neki način podrazumijevaju kretanje i djelovanje robota u radnom prostoru. Očekuje se da autonoman robot ima sposobnost planiranja i izvršavanja zadataka, koji u općenitom slučaju mogu zahtijevati i kretanje u nestacionarnim i nekarakteriziranim uvjetima. Zato ne iznenađuje činjenica da je problem planiranja kretanja privukao pažnju istraživačke zajednice već u ranoj fazi razvoja robotike. Podatak da danas još ne postoji robot koji se samostalno kreće ulicom ukazuje na težinu rješavanja ovog problema u uvjetima kada se okolina značajno mijenja u realnom vremenu.

Osnovna podjela pri planiranju kretanja odnosi se na stupanj poznavanja okoline u kojoj se kretanje izvodi. Kod klasičnog planiranja temeljenog na *modelu okoline* pretpostavlja se da sustav planiranja ima unaprijed učitanu mapu prostora. Kod planiranja temeljenog na *senzorskoj slici* okolina je nepoznata, a vođenje robota temelji se samo na očitanju sa senzorskih ulaza, bez konstruiranja mape prostora. Kod stvarnih se robotskih sustava ova dva pristupa mogu kombinirati i često se kombiniraju jer razne zadaće postavljene robotu mogu uzrokovati kontakt s okolinom.

S ciljem konzistentnog korištenja terminologije, želi se ovdje definirati sljedeće pojmove: *radni prostor* je fizički prostor u kojem se robot nalazi i djeluje. *Konfiguracija* robota predstavlja položaj svih točaka robota u radnom prostoru. Skup svih konfiguracija čini *konfiguracijski prostor* robota, u literaturi poznat pod engleskom kraticom *cspace* robota, oznake $\mathcal{C}$. Najmanji broj neovisnih parametara kojima je moguće u potpunosti odrediti konfiguraciju naziva se brojem *stupnjeva slobode gibanja* (*SSG*). *Slobodni prostor* odnosi se na područja radnog prostora koja nisu zauzeta preprekama ili područja u $\mathcal{C}$ u kojima robot ne kolidira s preprekom, što se u literaturi naziva i slobodni *cspace*, ili $\mathcal{C}_f$. Putanja je krivulja u $\mathcal{C}$. Putanja je prihvatljiva ako ne rezultira kolizijama između robota i prepreka. Ukoliko se nizu točaka s putanje dodijeli dimenzija vremena, takva se struktura naziva *trajektorijom*.

Ovi koncepti koriste se za formuliranje *osnovnog problema planiranja kretanja* [6]:

„Neka je $\mathcal{A}$ kruto tijelo – robot – koji se miče u Euklidskom prostoru $\mathcal{W}$, nazvanom radnim prostorom $\mathbf{R}^N$, pri čemu je $N = 2$ ili 3.

Neka su $\mathcal{B}_1,\ldots,\mathcal{B}_q$ nepomični kruti objekti raspoređeni u $\mathcal{W}$. $\mathcal{B}_i$ se naziva preprekom.Pretpostavlja se da su u danom trenutku poznati geometrija objekata $\mathcal{A}$, $\mathcal{B}_1,\ldots,\mathcal{B}_q$, kao i položaji $\mathcal{B}_i$ u $\mathcal{W}$. Dalje se pretpostavlja da ne postoje kinematska ograničenja kretanja $\mathcal{A}$.

Problem je: za danu početnu poziciju i orijentaciju i krajnju poziciju i orijentaciju $\mathcal{A}$ u $\mathcal{W}$, treba generirati putanju p koja određuje neprekinuti niz pozicija i orijentacija $\mathcal{A}$, izbjegavajući dodire s $\mathcal{B}_i$, počevši s početnom pozicijom i orijentacijom i završavajući s krajnjom pozicijom i orijentacijom.“

Prikazani se problem svodi na geometrijsko planiranje putanje za jedno kruto tijelo. Iako je pojednostavljen, ovaj problem je težak za rješavanje. Postoje mnoga proširenja ovog izvornog problema, kao i pristupi za rješavanje ovih proširenja. Prikazani je problem *statičke prirode*: informacije o položajima svih prepreka prisutne su na početku planiranja i ne mijenjaju se u bilo kojem sljedećem trenutku. Kod *dinamičke* varijante ovog problema, dodatne informacije o preprekama dobivaju se tijekom faze planiranja, npr. uslijed interakcije sa senzorima. U slučaju kada se okolina mijenja ili se prepreke kreću, problem postaje *vremenski varijabilan*. Kada se u radnom prostoru nalazi više robota, problem se naziva *problemom višestrukog kretanja*. Ukoliko objekti u radnom prostoru mogu mijenjati oblik, problem postaje *konformabilan*. Važnu podklasu konformabilnih problema čine roboti koji se sastoje od više krutih segmenata povezanih zglobovima, rotacijskog ili translacijskog karaktera i takvi se roboti nazivaju *artikuliranim* robotima. Robotski sustav može imati inherentne restrikcije na dostupno područje, što vodi na problem planiranja *s ograničenjima*. Kinematske veze ograničavaju kretanje robota ili njegovih dijelova. Ukoliko se neka ograničenja mogu ukloniti reparametrizacijom konfiguracije, takva se ograničenja nazivaju holonomnim ograničenjima. Holonomna ograničenja reduciraju broj parametara koji su potrebni za specifikaciju konfiguracije. Ona fundamentalno ne mijenjaju prirodu problema, nego samo smanjuju dimnezionalnsot prostora $\mathcal{C}$. Ograničenje koje je neintegrabilno s obzirom na parametre konfiguracije i njihove derivacije (parametre brzine) naziva se neholonomnim ograničenjem.

Računska složenost problema planiranja putanje ograničila je značajno razvoj praktičnih algoritama i njihovu primjenu [7]. S ciljem razumijevanja kompleksnosti problema planiranja kretanja i motivacije za primjenu heurističkih algoritama, potrebno je prikazati nekoliko teoretskih analiza problema planiranja kretanja. U takvim analizama, činjenica da se robot mora nalaziti u slobodnom dijelu radnog prostora reprezentirana je skupom jednakosti i nejednakosti, kojima se osigurava da bilo koji dio robota ne dodiruje ili presijeca bilo koju

značajku prepreke. Prostor $\mathcal{C}_f$ robota s k stupnjeva slobode gibanja, *SSG* se tada reprezentira kao podskup $\mathbf{R}^k$ definiranog Booleovim kombinacijama ograničenja postavljenih na robota. Takav se skup naziva semialgebarskim skupom. Veličina ovog skupa jednaka je broju jednakosti i nejednakosti i naziva se *geometrijskom* ili *kombinatornom* složenošću promatranog problema planiranja kretanja. Poznato je da je kombinatorna složenost prostora $\mathcal{C}_f$ robota s k *SSG* uz n geometrijskih ograničenja jednaka $\Theta\left(n^k\right)$. Obzirom da je potrebno izračunati samo povezanu komponentu u $\mathcal{C}_f$ koja sadrži početni položaj robota i uz neke prirodne geometrijske pretpostavke, složenost se reducira na $\Theta\left(n^{k-1}\right)$.

U literaturi se nalazi veći broj egzaktnih ne-heurističkih algoritama s određenim međama složenosti za različite probleme planiranja kretanja, no ti algoritmi u velikoj većini slučajeva nisu prikladni za praktičnu upotrebu [8]. Za sustave s većim brojem stupnjeva slobode gibanja, egzaktni algoritmi postaju u praksi vrlo neefikasni [9]. Ovi rezultati ukazuju na potrebu za primjenom aproksimacijskih ili heurističkih metoda kod praktične implementacije na stvarnim robotskim sustavima [10].

1.2 Tradicionalne metode planiranja kretanja

Tradicionalni pristupi planiranju kretanja mogu se podijeliti u tri osnovne skupine [6]: *metode mape putova, metode dijeljenja prostora*, i *metode potencijalnog polja*.

Neka je $\mathcal{C}$ oznaka skupa svih konfiguracija robota, a $\mathcal{C}_f$ slobodan konfiguracijski prostor, odnosno podskup skupa $\mathcal{C}$ gdje robot ne kolidira s preprekama. Osnovna ideja metode mape putova je pronalaženje mreže povezane jednodimenzionalnim krivuljama – putovima koji se nalaze u $\mathcal{C}_f$. Nakon konstrukcije mape putova $\mathcal{R}$, planiranje putanje svodi se na povezivanje početne i konačne konfiguracije u $\mathcal{R}$, i pretrage $\mathcal{R}$ za pronalaženje putanje.

Metoda dijeljenja prostora odvija se u tri koraka. U prvom koraku slobodan prostor $\mathcal{C}_f$ dijeli se na skup podprostora – ćelija, koje se ne preklapaju, a čija unija je upravo $\mathcal{C}_f$. U drugom se koraku konstruira graf povezanosti koji predstavlja veze između ćelija te se graf povezanosti pretražuje. Ukoliko je pretraga uspješna, izlaz je odgovarajući niz ćelija, koji se

naziva *kanalom*, kojim su povezane ćelija koja sadrži početnu konfiguraciju i ćelija koja sadrži konačnu konfiguraciju. Putanja se jednostavno rekonstruira iz ovog niza ćelija.

Kako bi se problem planiranja putanje uspješno riješio dvjema izloženim metodama, podrazumijeva se raspolaganje podacima o promatranom radnom prostoru unaprijed. Ovi se podaci mogu upotrijebiti za rješavanje jednog ili više problema planiranja, uz promjenu početne i konačne konfiguracije robota. Značajno i veoma poželjno svojstvo ove dvije metode je da u pravilu pokazuju svojstvo kompletnosti, odnosno, ukoliko postoji putanja koja povezuje početnu i konačnu konfiguraciju, ona će biti pronađena. Nedostatci su:

- izražena potreba za računalnim resursima, kako memorijskim tako i vremenskim i
- metode ne pokazuju dobra svojstva kod neholonomnih robota.

Kod metode potencijalnih polja robot se tretira kao čestica koja se nalazi pod utjecajem *umjetnog polja potencijala* $\mathbf{U}$. Deformacijom tog polja reflektira se narav slobodnog konfiguracijskog prostora $\mathcal{C}_f$. Potencijalno polje uobičajeno se definira kao funkcija $\mathcal{C} \rightarrow \mathbb{R}$ što predstavlja ponderiranu sumu *privlačnog* potencijala koji privlači robota prema krajnjoj konfiguraciji i *odbojnog* potencijala koji odbija robota od prepreka. Planiranje kretanja izvodi se ponavljanim izračunom umjetne sile pod čijim se djelovanjem robot kreće.

Osnovni nedostatak ove metode je to što robot može ostati zarobljen u lokalnom minimumu potencijalnog polja, odnosno, može se dogoditi da robot dosegne konfiguraciju S pri kojoj je ponderirana suma svih potencijala nul-vektor. Kako bi se ovakve pojave svele na najmanju moguću mjeru, potrebno je odabrati odgovarajuće potencijalne funkcije s minimalnim brojem lokalnih optimuma te primijeniti metode izlaženja iz suboptimuma. Danas postoje implementirane metode planiranja temeljene na potencijalnim poljima kojima je moguće planirati putanje za robote s većim brojem stupnjeva slobode, iako je problem pronalaska funkcija bez lokalnih minimuma izuzetno težak [11].

1.3 Alternativni pristupi planiranju kretanja

Rodney Brooks [12] 1989. godine postavlja pitanje opravdanosti koncepta *osjeti-interpretiraj-planiraj-djeluj*. Predlaže alternativne upravljačke obrasce temeljene na bihevioristićkom pristupu. Osnovne karakteristike ovog pristupa su da je modeliranje i reprezentacija vanjskog svijeta u robotskoj memoriji svedena na nužnu mjeru. Robot nema

spoznaje o tome po kakvoj se podlozi kreće, koje su dimenzije prepreke koju treba zaobići itd. Robot se ponaša kao reaktivni sustav oslanjajući se na trenutnu senzorsku sliku kako bi planirao buduće djelovanje [13]. Brooksovi roboti pokretani nogama pokazali su dotada nezabilježenu brzinu kretanja nalik insektima.

Problem planiranja putanje može se formulirati kao optimizacijski problem pri čemu je potrebno funkciju koja opisuje kretanje robota optimirati s obzirom na jedan ili više kriterija. Obzirom da optimizacijski kriteriji mogu biti suprotstavljeni, problem planiranja poprima obilježja *višekriterijskog optimizacijskog problema*, pri čemu često nije moguće pronaći jedinstveno globalno-optimalno rješenje [14], [15].

Evolucijski algoritmi su stohastička metoda pretrage prostora stanja, a pokazali su se vrlo pouzdanima pri optimizaciji kompleksnih multimodalnih domena [16]. Metode nastale na temelju apstrakcije ponašanja socijalnih insekata, u prvom redu mrava i pčela također se uspješno primjenjuju u domeni planiranja kretanja. Ove metode nazivaju se skupnim imenom *metodama inteligencije roja čestica* [17].

Sljedeći pristup planiranju putanje koji problem promatra u svjetlu optimiranja je primjenom teorije optimalnog upravljanja. Rješenje problema optimalnog upravljanja zahtijeva rješenje tzv. *two-point boundary value* problema koji je analitički rješiv za vrlo ograničen skup problema. Predlažu se numeričke metode rješavanja ovog problema temeljene na metodama linearnog programiranja i diskretizaciji prostora stanja i u novije vrijeme metode temeljene na gradijentu funkcije cilja. Pri tome se eliminiraju ograničenja i rubni uvjeti te se zamjenjuju funkcijama kazne, čime se rješavanje izloženog problema pojednostavljuje [18].

Metoda koja se odlikuje mogućnošću obrade velike količine podataka uz koje je prisutan šum je *neizrazita logika*. Naime, općenito senzorska slika robota za dvije istovjetne situacije nikada nije ista kod realnih sustava. Prisutna su ili kašnjenja, kod digitalnih ulazno-izlaznih i komunikacijskih modula, ili se, kod analognih očitanja, vrijednosti očitanja razlikuju u određenom postotku. Metode neizrazite logike koriste se na više načina, najčešće za određivanje pravila ponašanja robota u obliku *ako-onda* pravila [19].

Jedan od pristupa planiranju putanje temeljenom na metodama umjetne inteligencije je primjenom umjetnih neuronskih mreža. Pri tome se mreža najčešće koristi tako da ulazni sloj neurona prima senzorsku sliku robota. Izlazni sloj neurona uparen je s izvršnim elementima robota. Ova metoda odvija se u dvije faze. Prva faza je faza učenja, u kojoj je vrlo bitno

izabrati odgovarajući skup za učenje, najčešće su to karakteristična stanja u kojima se očekuje da će se robot tijekom djelovanja naći. U drugoj fazi, mreža je sposobna dati odgovor za poznate slučajeve iz skupa za učenje, ali i uslijed poželjnog svojstva *generalizacije* rješavati situacije koje su nepoznate, odnosno nisu bile prisutne u skupu za učenje [20].

U literaturi se razmatraju i drugi pristupi rješavanju problema planiranja putanja, međutim oni se većinom svode na neke od opisanih pristupa. Činjenica da danas nema robota koji se autonomno kreću cestom govori o složenosti problema planiranja putanje. U sljedećem odjeljku formulirati će se problem planiranja putanje kada više robota dijeli radni prostor.

1.4 Više robota u radnom prostoru

Proširenje osnovnog problema planiranja putanje je slučaj kada u radnom prostoru zajednički djeluju najmanje dva robota, ili jedan robot i najmanje jedna pomična prepreka. Kod ove vrste problema, koja se naziva *dinamičkim problemom planiranja kretanja* nije primjenjiv pristup planiranja statičke, vremenski neovisne putanje. Potrebno je u obzir uzeti i vremensku komponentu, koja je ireverzibilna. Dodatno se model može približiti realnoj situaciji tako da se postavljaju odgovarajuća ograničenja na maksimalnu brzinu i akceleraciju robota.

Pri razmatranju problema dinamičkog planiranja razlikuju se dvije inačice problema:

- upravljiv je jedan ili više robota uz prisutnost neupravljivih dinamičkih objekata – prepreka,
- u radnom prostoru se nalaze samo roboti te su svi upravljivi.

U ovoj disertaciji razmatrana je druga inačica, odnosno u radnom prostoru ne nalaze se neupravljive dinamičke prepreke. Točnije, razmatraju se takvi roboti koju su konačnih dimenzija, sastavljeni od parova pomičnih elemenata povezanih rotacijskim zglobovima.

Općenito se problem višerobotskog planiranja putanje može formulirati ovako: u radnom prostoru nalazi se p robota, $\mathcal{A}_i$, i=1,…,p, koji se kreću u istom radnom prostoru $\mathcal{W}$=R^N, N=2 ili 3, između statičkih prepreka $\mathcal{B}_j$, j=1,…,q. Roboti se kreću neovisno jedan o drugome, ali ne mogu se u istom trenutku nalaziti na istom mjestu. Zadatak je pronaći putanju bez kolizija za svakog robota od njegove početne do konačne konfiguracije.

Postoje dva osnovna pristupa rješavanju izloženog problema: *centralizirane* metode i *decentralizirane* metode.

Kod centraliziranog pristupa, svi roboti grupiraju se zajedno tvoreći jednog jedinstvenog, kompozitnog robota. Nakon toga, problem se svodi na planiranje gibanja u prostoru koji sadrži jednog robota. Problem je što u ovom slučaju kompozitni robot uobičajeno ima puno stupnjeva slobode gibanja [6], [21]. To je značajno ograničenje jer je pokazano da kompleksnost planiranja putanje raste eksponencijalno s brojem stupnjeva slobode gibanja. Razvijene su specijalne, manje vremenski zahtjevne centralizirane metode planiranja putanje za višerobotske sustave, za slučaj kada zasićenost radnog prostora nije velika [22].



Slika 1-3. Primjer početne (lijevo) i konačne (desno) konfiguracije za koje decentralizirani pristup ne daje zadovoljavajuće rješenje.

Složenost centraliziranog pristupa dovela je do razvoja decentraliziranih metoda planiranja putanje za višerobotske sustave [23], [24]. Kod decentraliziranog pristupa se potpunost rješenja svjesno žrtvuje s ciljem redukcije složenosti problema. Kod osnovnog decentraliziranog pristupa, planiranje se putanje provodi u dvije razdvojene faze. U prvoj se fazi za svakog robota izvodi planiranje uzimajući u obzir samo statičke prepreke iz radnog prostora, izuzimajući potpuno druge robote. Samo izbjegavanje kolizije rješava se u drugoj fazi, modulacijom brzine robota. Brzina robota mijenja se tako, da se robot koji se kreće po putanji dobivenoj u prvoj fazi, ne dodiruje u nijednom trenutku s drugim robotima. Ova metoda je inherentno nepotpuna, iako metode primijenjene u prvoj i drugoj fazi mogu biti potpune, što ima za posljedicu da za određene početne i konačne uvjete, za koje postoji rješenje u obliku slobodne trajektorije za p robota, nije ovim metodama moguće pronaći to rješenje [25].

Situacija prikazana na slici [Slika 1-3] općenito je kompliciran slučaj kod kojega roboti moraju značajno odstupiti od optimalnih trajektorija (najčešći je kriterij dužina putanje). Decentralizirani pristup rješavanju ovog problema najvjerojatnije nije u mogućnosti

pronaći rješenje uslijed razdvojenosti faza planiranja i faze modulacije brzine. Detaljna analiza analognog scenarija za slučaj kada su roboti sastavljeni od segmenata povezanih rotacijskim zglobovima dana je u poglavlju 3.2.7.

1.5 Alternativne metode planiranja putanje u višerobotskim sustavima

Obzirom na složenost problema planiranja putanje općenito, a posebice u slučaju kada okolina sadrži više pomičnih objekata, intenzivno se traže nove metode koje bi bile u stanju učinkovito rješavati problem planiranja putanje. Razvijene su metode temeljene na evolucijskim algoritmima [26], [27], [28]. Pretežito je primjena evolucijskih algoritama usmjerena na pronalaženje referentnih točaka spline-ova koji određuju trajektoriju po kojoj se kreće referentna točka na izvršnom članu robota.

Kod metoda temeljenih na neizrazitoj (eng. *fuzzy*) logici, prikazano je u [29] generiranje složenog koordiniranog rada između dva manipulatora s ciljem zajedničkog nošenja tereta.

U radu [21] prikazana je kombinirana metoda koja dijelom objedinjuje metode decentraliziranog i centraliziranog pristupa planiranju putanje za višerobotske sustave. Osnovna je ideja da se istodobno, za vrijeme traženja putanje jednog robota, traže i konfiguracije odnosno položaji robota na putanji za robote koje je putanja već pronađena. Motivacija je da roboti koji već imaju definirane putanje mogu modulacijom svoje brzine na tim putanjama olakšati pronalaženje putanje koja ne rezultira kolizijom za robota za kojeg se trenutno traži putanja. Prikazane su usporedbe klasičnih, decentraliziranih i centraliziranih metoda s novorazvijenom i pokazana su vrlo dobra svojstva u smislu skraćenja vremena konvergencije, uz minimalno žrtvovanje potpunosti algoritma.

U [30] prikazani su preliminarni rezultati temeljeni na metodi *konstrukcijske geometrije* za planiranje putanje dvije robotske ruke koje dijele radni prostor. Korišten je dvofazni evolucijski algoritam kojim se kombinira pretraga slobodnog prostora i istovremeno traži cilj u prethodno pretraženom području. Razvijene su metode temeljene na uzorkovanju konfiguracijskog prostora $\mathcal{C}$ no obzirom da problem planiranja putanje nije traktabilan, ove su metode ograničeno primjenjive [31], [32]. Metoda inteligencije roja čestica (eng. *Particle Swarm Optimization, PSO*) uspješno je u [33] primijenjena na držanje formacije grupe mobilnih robota. Predložen je upravljački algoritam temeljen na Lyapunovim funkcijama i

teoriji grafova za kojega je pokazano da uz određene uvjete, čuva formaciju od početne do krajnje točke u prostoru.

Obzirom da je prostor pretrage prostora stanja u slučaju planiranja putanje vrlo velik, pokazalo se da enumeracijske metode, temeljene na diskretizaciji unije konfiguracijskog prostora svih p robota nisu prikladne [28]. U radu [34] kombinirane su metode umjetnih neuronskih mreža i genetskog algoritma za rješavanje planiranja putanje za pet karakterističnih scenarija. Pri tome je neuronska mreža u jednom slučaju kao metodu učenja imala povratno rasprostiranje greške, a u drugom je učenje izvođeno genetskim algoritmom. Genetski algoritam u prikazanom se radu pokazao kao prikladniji, u prvom redu jednostavniji za implementaciju i vremenski efikasniji izbor metode učenja.

Relativno dobri rezultati planiranja postižu se primjenom metoda temeljenih na stohastičkom uzorkovanju konfiguracijskog prostora $\mathcal{C}$. Najraširenija je probabilistička metoda putova (eng. *Probabilistic Roadmap Method, PRM*). Metoda se temelji na slučajnom uzorkovanju $\mathcal{C}$ te memoriranju dobrih prolaznih točaka. Bolje su one prolazne točke koje rezultiraju putanjom slobodnom od kolizija te one koje rezultiraju kraćom putanjom.

Obzirom na probleme koji su izloženi i na nepostojanje općeprihvaćene metode za planiranje putanje u višerobotskim sustavima, a imajući u vidu inherentnu distribuiranost procesa, predlaže se metoda temeljena na paralelnim evolucijskim algoritmima – koevolucijskom algoritmu. Ova metoda, iako prema gornjem razmatranju spada u područje centraliziranih metoda, jer je konstantno poznato stanje svih robota u sustavu, koristi se činjenicom da su prostori stanja dva robota vezani. Ti se prostori zajednički pretražuju i zapravo prilagođavaju jedan drugom. Dodatno, određene se operacije mogu izvoditi distribuirano i potpuno neovisno, poput inicijalizacije i dijelom evaluacije dviju koevoluirajućih populacija, dok se svi evolucijski operatori mogu izvoditi paralelno. Time se značajno štedi vrijeme potrebno za pronalaženje nizova konfiguracija lijevog i desnog robota. Osnovni nedostatak ove metode je što počiva na stohastičkim procesima čime se ne može garantirati optimalnost pronađenog rješenja.

U sljedećem se poglavlju objašnjavaju evolucijski algoritmi, metode, prirodne inspiracije, klase evolucijskih algoritama, njihova povijest, teoretske osnove kao i prednosti i ograničenja.

1.6 Cilj i hipoteza rada

Cilj rada je razvoj generičkog upravljačkog sustava sastavljenog od dvaju robota $2R$[ii] konfiguracije i adaptivnog koevolucijskog algoritma kojim je moguće efikasno planiranje putanje za dva robota u uvjetima kada roboti dijele radni prostor. U slučaju kada oba robota imaju pristup određenom dijelu radnog prostora, oni ulaskom u taj prostor predstavljaju dinamičku prepreku jedan drugome.

Zahtjevi koji se pri tome postavljaju na trajektorije pronađene koevolucijskim algoritmom su sljedeći:

- trajektorije trebaju biti izvedive na realnom robotskom sustavu,
- trajektorije ne smiju rezultirati kolizijom bilo kojeg segmenta lijevog robota sa segmentom desnog robota ili sa statičkom preprekom u radnom prostoru,
- trajektorije moraju dovoljno dobro zadovoljavati ostale kriterije.

U radu se razmatra sustav sastavljen od dva robota koji su upravljani zasebnim upravljačkim računalima te ne dijele memorijske registre niti imaju zajedničko procesorsko vrijeme. Pretpostavlja se da je moguće istodobno planiranje putanje za dva robota primjenom paralelnih, dijelom neovisnih evolucijskih algoritama – koevolucijskih algoritama. Pretpostavlja se da je primjenom koevolucijskih algoritama moguće efikasno paralelno pretraživanje vezanih prostora stanja dvaju robota.

Nadalje, obzirom na inherentnu distribuiranost izloženog problema, predloženi algoritam ima poželjna svojstva istodobnog izvođenja računskih operacija poput inicijalizacije i dijelom evaluacije populacija potencijalnih rješenja, te svih evolucijskih operatora na dvama upravljačkim računalima. Informacije između upravljačkih računala potrebno je izmijeniti samo u trenutku evaluacije kolizije. Istodobno izvođenje ovih procesa osigurava uštedu procesorskog vremena.

Kriteriji vrednovanja trajektorije su sljedeći:

- broj kolizija s drugim robotom ili statičkim preprekama u radnom prostoru,
- ukupni kut rotacije u zglobovima robota,
- ukupna duljina trajektorije i

[ii] $2R$ – konfiguracija robota koji se sastoji od dva rotacijska zgloba, poput robota SCARA (eng. *selective compliant assembly robot arm*) konfiguracije u horizontalnoj ravnini.

- neravnomjernost profila brzine referentne točke na izvršnom članku robota.

Rješenje izloženog problema dobiva se u obliku diskretnih kutova zakreta svakog zgloba lijevog i desnog robota. Interpolacijom kroz referentne kutove zakreta polinomom reda *n* veće od dva, dolazi se do kontinuirane vremenske ovisnosti zakreta svakog zgloba. Iz ove se ovisnosti rješavanjem direktnog kinematičkog problema dolazi do vremenske ovisnosti pomaka svake točke lijevog i desnog robota.

Na temelju iznesenog, može se postaviti sljedeća hipoteza:

Pretpostavlja se da je razvojem adaptivnog koevolucijskog algoritma zasnovanog na promjenjivim parametrima ovisnima o trenutnoj distribuciji populacija, traženju i identificiranju najboljih kolaboratora, te modificiranim postupcima među-populacijske evaluacije, moguće istodobno planiranje putanja više agenata koji dijele radni prostor. Također, pretpostavlja se da je modificiranim koevolucijskim algoritmom moguće osigurati dovoljno dobro ispunjavanje zadanih kriterija i osigurati pouzdani rad višeagentskog sustava u realnoj okolini.

1.7 Metodologija i plan istraživanja

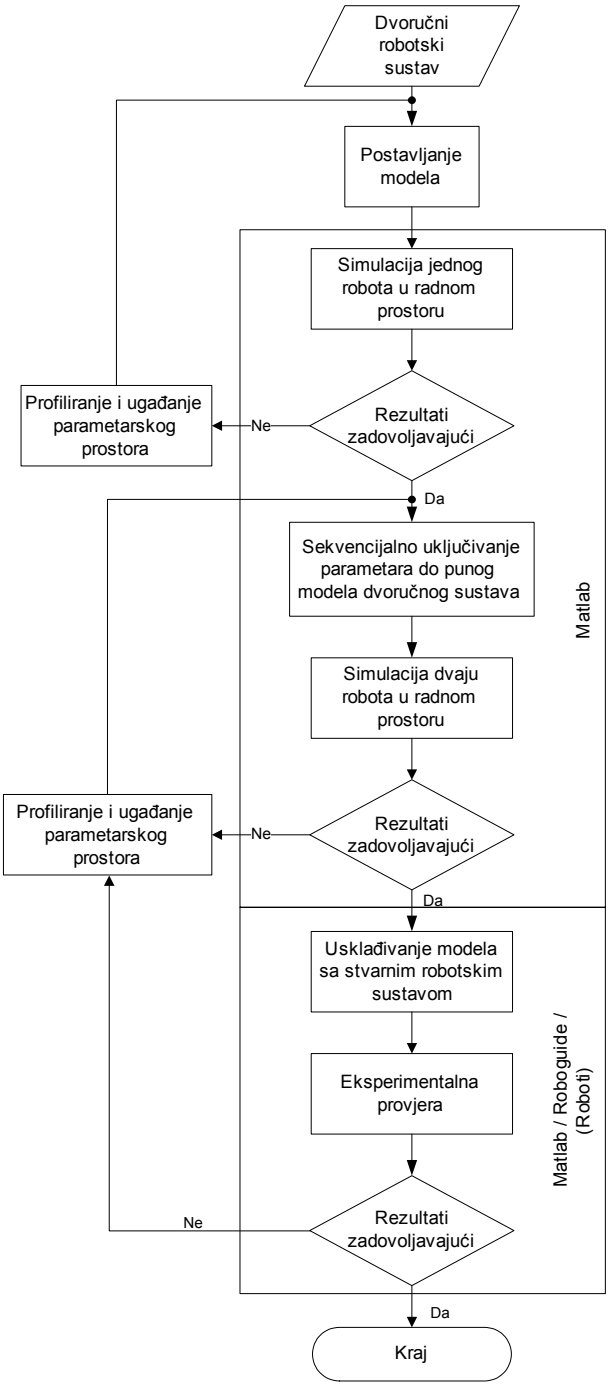
Kako bi se navedena hipoteza potvrdila, metodologija istraživanja sadržavat će definiranje, analizu i postavljanje odgovarajućeg modela dvoručnog robota s potrebnim brojem stupnjeva slobode gibanja. Validacija i verifikacija modela izvest će se simulacijom na seriji prethodnih jednostavnijih problema. Sekvencijalnim uključivanjem optimizacijskih kriterija postupno će se implementirati puni model dvoručnog robota.

Model će biti simuliran u programskom okružju MatLab, uz opsežno ispitivanje za različite početne i konačne uvjete dvoručne konfiguracije. Kao rezultat simulacije bit će dobivene koordinate robota zapisane u obliku kutova zakreta odgovarajućih zglobova u zadanom vremenskom intervalu. Profinjavanjem vremenske diskretizacije omogućit će se dobivanje dodatnih koordinata robota, no ovaj pristup je nepovoljan s aspekta koevolucijskog algoritma jer se prostor pretrage značajno povećava. Stoga će se, nakon proračuna referentnog niza točaka, interpolacijom polinomom reda *n* veće od dva, kroz referentne točke dobiti raspodjela zakreta u zglobovima u vremenski neprekinutom intervalu. Obzirom da ovako dobiveni polinom predstavlja raspodjelu kuta zakreta u vremenu, potrebno je, provjerom

odgovarajuće derivacije tog polinoma osigurati neprekinutost profila brzine, ubrzanja, te omeđeni iznos trzaja dopušten fizičkim ograničenjima robota.

Dodatna provjera modela izvršit će se u realnom fizičkom simulacijskom okružju Roboguide u kojem je omogućena vrlo detaljna i vjerna analiza u realnim uvjetima vremenskog kašnjenja na razini procesorske i fizičke vremenske domene.

Određeni segmenti istraživanja implementirat će se na realnom robotskom sustavu sastavljenom od dvaju robota 2*R* konfiguracije.



Slika 1-4. Dijagram toka istraživanja.

Istraživanja će bit provedena po sljedećim fazama, [Slika 1-4]: prva faza bit će usmjerena na modeliranje i simulaciju rada dvoručnog robota. Nakon inicijalnih ispitivanja,

predložit će se u drugoj fazi modifikacije koevolucijskih algoritama usmjerene prema osiguravanju konvergencije i ubrzavanju rada algoritma. Treća faza istraživanja obuhvatit će eksperimentalni dio i verifikaciju predložene metodologije. Nakon implementacije punog modela dvoručnog robota i pripadajućeg koevolucijskog upravljačkog algoritma, objedinit će se spoznaje iz prethodnih faza. Ove će spoznaje poslužiti za daljnje profiliranje parametarskog prostora, te konačno, u četvrtoj fazi punu implementaciju predložene metodologije na dvoručnoj robotskoj konfiguraciji.

1.8 Očekivani znanstveni doprinos

Znanstveni doprinos predloženog istraživanja sastoji se u razvoju modificiranog evolucijskog algoritma kojime je moguće vremenski efikasno planiranje optimizirane putanje za svaku ruku dvoručnog robotskog sustava. Na temelju iscrpnih simulacijskih provjera utvrdit će se priroda nelinearnih veza između pojedinih parametara evolucijskih algoritama što će pridonijeti boljem razumijevanju načina rada evolucijskih algoritama općenito. Daljnji je doprinos u razvoju generičkog tehničkog sustava sastavljenog od dvaju robota upravljanih razvijenim algoritmom u uvjetima kada dva robota dijele radni prostor i zajednički rješavaju zadani problem, što ih onda čini dvoručnim robotskim sustavom.

1.9 Struktura disertacije

Disertacija je strukturirana kako slijedi:

- U prvom, uvodnom poglavlju, problem planiranja kretanja stavljen je u konetkst robotskih istraživanja koja se danas provode, kako za slučaj uslužne, tako i za slučaj industrijske robotike. Prikazane su značajnije metode planiranja putanje te su istaknute prednosti i nedostatci ovih metoda. Na temelju ovih spoznaja, formulirana je hipoteza rada, predstavljen je plan istraživanja, te je izložen znanstveni doprinos disertacije.
- Drugo poglavlje, *Evolucijski algoritmi* sadrži pregled područja evolucijskog računarstva. Polazi se od definicija osnovnih pojmova, populacije, križanja, selekcije. Prikazane su teoretske osnove rada evolucijskih algoritama te njihova analiza s obzirom na računsku složenost. Objašnjava se priroda procesa

optimizacije primjenom evolucijskih algoritama te se ističu njihove prednosti, kao i nedostatci u odnosu na druge optimizacijske metode.

- *Razvoj višeagentskog modela upravljanja dvoručnim robotom* treće je poglavlje u koje se definiraju pojmovi *agenta*, *višeagentskog sustava* i povezuje ih se s promatranim dvoručnim robotskim sustavom. Postavlja se model 2*R* robota te se razvija i prikazuje simulacijsko okružje u kojem se izvodi planiranje kretanja. Komponente modela uvode se sekvencijalno sve do punog modela dvoručnog robota 2*R* konfiguracije. Ovaj se model ispituje i prikazuju se rezultati simulacije uz različite početne i konačne uvjete.
- Poglavlje četiri bavi se validacijom i verifikacijom predložene metodologije u realnom fizičkom simulacijskom okružju Roboguide Fanuc roboticsa. Objašnjavaju se karakteristike sustava Roboguide. Rezultati dobiveni predloženim evolucijskim algoritmom predstavljaju ulaze u realni robotski sustav u vidu zakreta odgovarajućeg stupnja slobode robota. Diskutira se odstupanje modela primijenjenog u drugom poglavlju od stvarnih robote i predlažu se metode za eliminaciju ovog odstupanja. Prikazuju se rezultati simulacije u sustavu Roboguide, kao i na dva realna robota Fanuc Lr Mate 200iC.
- Peto je poglavlje zaključno. U njemu se rekapituliraju polazne pretpostavke provedenog istraživanja i dovode u vezu s rezultatima dobivenim nizom simulacija i eksperimentima na realnom robotskom sustavu. Kritički se osvrće na ove rezultate te se predlažu smjerovi budućeg istraživanja temeljenog na spoznajama proizašlim iz provedenog istraživanja.
- Popis korištenih bibliografskih jedinica nalazi se u posljednjem, šetom poglavlju.

2 Evolucijski algoritmi

Evolucijsko računarstvo (eng. *Evolutionary Computation, EC*) temelji se na povezivanju biološki inspiriranog procesa evolucije sa znanosti o računalima i umjetnom inteligencijom. Pojednostavljeni Darwinov evolucijski model primjenjuje se u računarstvu s ciljem rješavanja različitih optimizacijskih problema [35], [36]. Evolucijski algoritmi primijenjeni su i s ciljem modeliranja različitih ne-darwinovskih modela evolucije i traženje odgovora na pitanje *što ako*. U kontekstu tehničke primjene evolucijskih algoritama, pokazali su se uspješnim u rješavanju teških optimizacijskih problema, a posebno je intenzivno njihovo istraživanje u području višekriterijskih optimizacijskih problema u posljednje vrijeme [37].

Rješavanje mnogih problema u računarstvu temelji se na pretrazi prostora stanja vrlo velikih dimenzija. Ilustrativan primjer je onaj pri primjeni evolucijskih algoritama za stvaranje proteina, pri čemu algoritam pretražuje enormno velik broj kombinacija amino-kiselinskih sekvenci s ciljem pronalaženja željenih svojstava. Radi ilustracije veličine ovog prostora, čitav ljudski genom sastoji se od jedne milijarde nukleotidnih baza. Na svakoj poziciji unutar tog lanca informacija može se teoretski pojaviti jedna od četiri nukleotidne baze. Jednostavnim izračunom dolazi se do $4^{1000000000}$ mogućih kombinacija. Ovaj broj je toliko velik da je zapravo izvan granica poimanja[iii] [38].

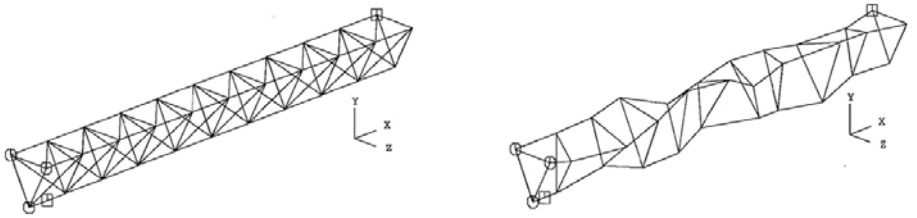
Mnogi problemi su takve prirode da je za njihovo uspješno rješavanje računarskim programom poželjno da je program *adaptivan*, odnosno da njegov učinak ostaje dobar i u promijenjenim uvjetima. Tipičan je problem iz ove klase upravljanje robotom temeljeno na

[iii] Radi ilustracije reda veličine ovog prostora stanja, potrebno je napomenuti da je danas broj 10^{120} prihvaćen kao neka vrsta granice složenosti, jer taj broj omeđuje gornju granicu broja mogućih slučajeva u svemiru do danas. Do tog broja dolazi se kao produkta 10^{80} stabilnih elementarnih čestica u svemiru i starosti svemira, mjerenoj u elementarnom vremenu, koje iznosi približno 10^{40}.

evolucijskim algoritmima, pri čemu robot zadaće izvršava u različitim ili nestacionarnim okolinama. Sljedeći primjer ove kategorije problema su računalna sučelja od kojih se zahtijeva prilagodba različitom krugu korisnika.

Rješavanje određenih problema zahtijeva od računarskih programa svojstvo *inovativnosti* – mogućnost da stvore uistinu novo i originalno rješenje. Ilustrativan primjer ovakve vrste problema koji je uspješno riješen genetskim algoritmom je optimiranje konstrukcije rešetkastog nosača. Zadatak je pronaći takav rešetkasti nosač koji će biti stabilan i posebice imati dobra svojstva prigušivanja vibracija. Naime, istraživanje je provođeno za NASA-u, pri čemu, uslijed specifičnosti atmosfere, viskozno prigušenje u svemiru je zanemarivo. U [39] prikazana je optimirana konstrukcija, koju karakteriziraju 20% bolja svojstva prigušivanja vibracija od do tada korištenih konvencionalnih rešetkastih nosača. Rešetkasti nosač optimiran genetskim algoritmom nije intuitivnog oblika, nije simetričan i na prvi pogled ne vidi se opravdanje njegovog izgleda. Upravo činjenica da genetski algoritam nije usmjeravan kriterijima estetičnosti, simetrije, naslijeđa itd., već isključivo kriterijem kvalitete rezultirala je rješenjem posebno dobrih svojstava. Rješenje dobiveno genetskim algoritmom i konvencionalan rešetkasti nosač prikazani su slikom [Slika 2-1].

Konačno, rješenja određenih problema su takve prirode da je vrlo teško ili nije uopće moguće napisati takav računalni kôd koji bi bio u stanju riješiti problem na zadovoljavajući način.



Slika 2-1. Primjer konvencionalnog rešetkastog nosača (lijevo) i konstrukcije dobivene genetskim algoritmom (desno).

Najpoznatija klasa iz ove skupine problema je stvaranje obilježja umjetne inteligencije. U ranoj fazi razvoja područja umjetne inteligencije vjerovalo se da će biti moguće na jednostavan i učinkovit način ugraditi u računalni program skup određenih pravila

i na taj način ostvariti inteligentno ponašanje umjetnih artefakata. Primjer ovog pristupa su ekspertni sustavi.

Danas se smatra da su pravila, odnosno obrasci koji se nalaze u pozadini inteligencije prevelikog stupnja složenosti da bi ih bilo moguće ugraditi u „top-down“ obliku. Nasuprot tome, kod „bottom-up“ pristupa, kodiraju se jednostavna pravila, a složene manifestacije, poput inteligencije, proizlaze iz tog skupa pravila. Konekcionistički pristup (proučavanje računalnih programa temeljeno na živčanom sustavu živih bića) jedan je primjer iz bottom-up skupine [40], dok je evolucijsko računarstvo drugi.

Biološka evolucija nepresušan je izvor inspiracija za rješavanje navedenih problema. Evolucija je zapravo metoda pretraživanja izuzetno velikog broja mogućih rješenja. U biologiji, taj broj je skup mogućih genskih sekvenci, a željena „rješenja“ su visoko kvalitetni organizmi – oni koji su sposobni za preživljavanje i reprodukciju u okolini koja ih okružuje. Upravo ti evolucijski mehanizmi izvor su inspiracija za računalne metode pretrage prostora stanja. Jasno je da su mehanizmi prirodne evolucije znatno kompleksniji od onih koji se primjenjuju kod evolucijskih algoritama.

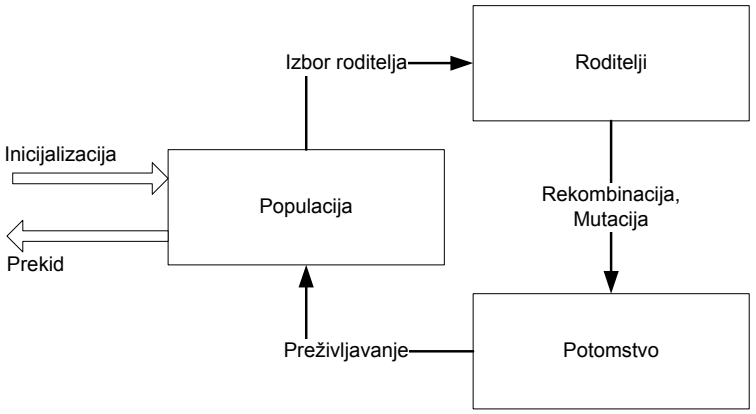
Osnovna tri mehanizma na kojima počivaju evolucijski algoritmi su pojednostavljeni mehanizmi *selekcije, rekombinacije* i *mutacije*. U prirodi, kriteriji uspješnosti kojima se evaluiraju pojedinci nestacionarni su i mijenjaju se kako se jedinke prilagođavaju svojoj okolini. Pretraga prostora rješenja u okolini promjenjivih karakteristika je upravo svojstvo koje se traži od adaptivnih računalnih programa. Sljedeća osobina evolucije je masovna paralelnost pri pretrazi prostora, evolucijski mehanizmi uvijek uključuju više jedinki, pa čak i veći broj vrsta koje su uzajamno mijenjaju i prilagođavaju.

Konačno, promatrano s aspekta zakonitosti visoke razine koje se nalaze u pozadini evolucijskih procesa, evoluciju karakteriziraju relativno jednostavni mehanizmi: vrste se razvijaju posredstvom slučajne promjene (putem križanja i rekombinacije), nakon čega slijedi selekcija koja omogućuje preživljavanje i reprodukciju kvalitetnijim, bolje prilagođenim jedinkama. Na taj se način genetski materijal kvalitetnijih pojedinaca prenosi na dolazeće generacije. Iako jednostavni, ovi mehanizmi u velikoj su mjeri odgovorni za stvaranje izuzetno složenih manifestacija u biosferi, uključujući ljudsku inteligenciju.

2.1 Osnovni evolucijski algoritam

Osnovni evolucijski algoritam izvodi se tako da se u prvom koraku po slučaju incijalizira početna populacija rješenja. U drugom koraku se ocjenjuje dobrota svakog člana populacije. Dobrota, f_i, pojedinca x_i ocjenjuje se funkcijom dobrote $\Phi : f_i = \Phi(x_i)$, pri čemu su pojedinci s većom dobrotom bolji. Funkcija dobrote omogućuje vođenje evolucijskog procesa u odgovarajućem smjeru, ona je zapravo reflektira svojstva okoline kojoj se pojedinci (potencijalna rješenja) trebaju prilagoditi. Danas nema općih smjernica za oblikovanje funkcije dobrote. Često se do odgovarajuće funkcije dobrote dolazi metodom pokušaja i pogreške, što može biti vremenski zahtjevno. Ovo je ujedno i najveća razlika između umjetne i prirodne evolucije kod koje nema definirane funkcije dobrote eksplicitno.

Nakon evaluacije, kreira se nova generacija populacije rješenja, primjenjujući selekcijske mehanizme i varijacijske operatore. Selekcijski mehanizmi osiguravaju da pojedinci iz populacije koji imaju veću vrijednost dobrote imaju veću šansu za preživljavanje i prijenos svog genetskog materijala u sljedeće generacije.



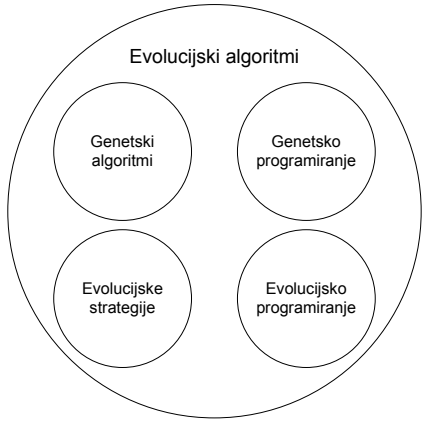
Slika 2-2. Dijagram toka općenitog evolucijskog algoritma.

Pseudo kôd općenitog evolucijskog algoritma ima sljedeći oblik:

```
POČETAK
        INICIJALIZIRAJ populaciju slučajnih kandidata;
        OCIJENI svakog kandidata
        PONAVLJAJ DO (UVJET PREKIDA ZADOVOLJEN)
                1 IZABERI roditelje;
                2 REKOMBINIRAJ parove roditelja;
                3 MUTIRAJ rezultirajuće potomke;
                4 OCIJENI nove kandidate;
                5 IZABERI jedinke za novu generaciju;
        prekid
KRAJ
```

Slika 2-3. Pseudo kôd osnovnog evolucijskog algoritma.

Evolucijski algoritmi skupno su ime za četiri kanonske klase algoritama, *genetske algoritme (GA)*, *evolucijske strategije (ES), evolucijsko programiranje (EP)* i *genetsko programiranje (GP)*. Različitost između pojedinih klasa većinom se odnosi na način kodiranja pojedinaca i na varijacijske operatore koji se primjenjuju s ciljem stvaranja novih pojedinaca.



Slika 2-4. Kanonski evolucijski algoritmi.

Pored klasa evolucijskih algoritama prikazanih slikom [Slika 2-4] postoje i, *memetički algoritmi*, *koevolucijski algoritmi*, *diferencijalna evolucija* i drugi, koji su izvedenice elemenata osnovne klase evolucijskih algoritama.

2.2 Metode kodiranja kod evolucijskih algoritama

Dva bitna koncepta evolucijskih algoritama odnose se na *fenotip* i *genotip*, odnosno na fenotipski $\mathcal{F}$ i genotipski prostor $\mathcal{G}$. Potrebno je odrediti metodu preslikavanja $\mathcal{F} \rightarrow \mathcal{G}$ i $\mathcal{G} \rightarrow \mathcal{F}$.

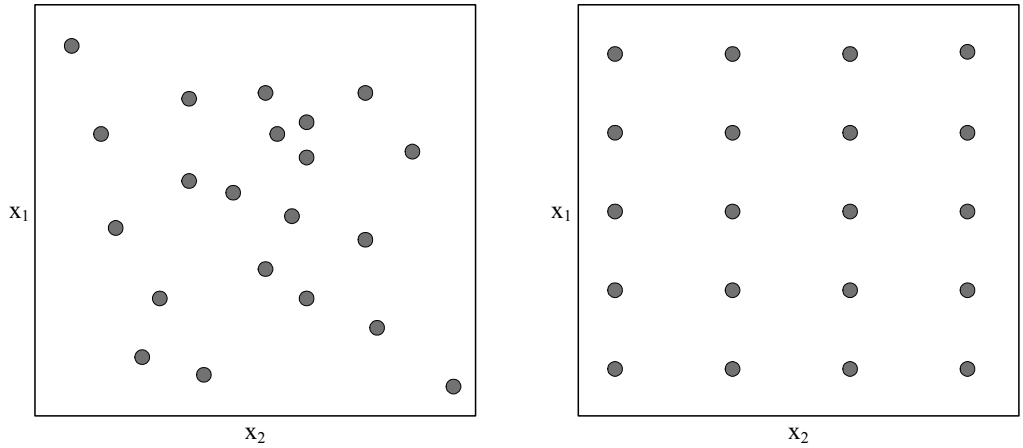
Kromosom A	1	0	1	1	0

Kromosom B	0,234	0,176	0,222	0,454

Slika 2-5. Primjer kromosoma kodiranog binarnim vrijednostima (kromosom A) i kodiranje realnim vrijednostima (kromosom B).

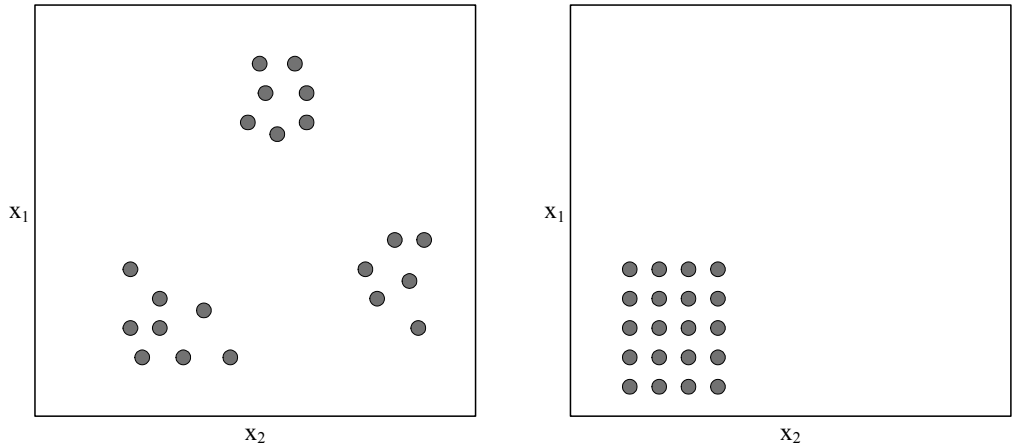
Jedno rješenje, jedan član populacije kodiran je najčešće kao niz znakova, polje, ili stablo i naziva se *kromosom* ili ponekad *gen*. Jednom kromosomu odgovara jedna točka u hiperprostoru rješenja, dok cijeloj populaciji odgovara skup točaka u istom hiperprostoru.

U pravilu se vrednovanje pojedinaca izvodi u prostoru $\mathcal{F}$, dok se ostali evolucijski operatori izvode u $\mathcal{G}$. Pri tome je *fenotip* ono što zapravo „vidimo“ – jedno od rješenja problema, dok je *genotip* samo prikaz tog rješenja prikladan za provođenje evolucijskih operatora. Duljina kromosoma ovisi o broju parametara koji su potrebni za rješavanje odgovarajućeg problema, a kao što je već rečeno, svaki kromosom će biti točka hiperprostora koji razapinju ti parametri.



a) Inicijalizacija po slučaju.

b) Mrežna inicijalizacija.



c) Inicijalizacija po slučaju uz distribuciju temeljenu na znanju.

d) Mrežna inicijalizacija uz distribuciju temeljenu na znanju.

Slika 2-6. Primjeri metoda inicijalizacije početne populacije rješenja. Kvadrati označavaju dvodimenzionalne prostore pretrage. Sivi krugovi predstavljaju fenotip pripadajućeg potencijalnog rješenja.

Iako su mogući mnogi načini kodiranja, najčešće se koristi *binarno* kodiranje i kodiranje *realnim vrijednostima*, kako je prikazano slikom [Slika 2-5]. Način dekodiranja

kromosoma ovisan je o problemu i ideji koja se nalazi iza izabranog kodiranja. Kao primjer, kromosom A sa slike [Slika 2-5] može se dekodirati u decimalni broj na sljedeći način:

$$x = 1\times 2^{-5} + 0\times 2^{-4} + 1\times 2^{-3} + 1\times 2^{-2} + 0\times 2^{-1} = 0,4625 \quad (1)$$

ili u cjelobrojnu vrijednost:

$$x = 1\times 2^{4} + 0\times 2^{3} + 1\times 2^{2} + 1\times 2^{1} + 0\times 2^{0} = 22, \quad (2)$$

ovisno o tome kako se implementira postupak dekodiranja.

Prilikom kodiranja i izbora odgovarajućeg načina dekodiranja, bitno je voditi računa o tome da globalni optimum bude dostupan iz prostora genotipa. Naime, kod kromosoma A, najveći x kojemu je moguće pristupiti je $x = 32$. Ukoliko se rješenje nalazi negdje u prostoru $x > 32$, tada primjenom ovog kodiranja, algoritam nikad neće konvergirati.

Inicijalizacija početne populacije u pravilu se izvodi stohastički, no, ukoliko je unaprijed dostupno ekspertno znanje o domeni koja se pretražuje, moguće je ugraditi ovo znanje u postupak inicijalizacije.

U primjeru prikazanom slikom [Slika 2-6] u slučajevima a) i b) nije dostupno znanje o domeni pretrage. U slučajevima c) i d) postoji određeno znanje o domeni, te je populacija inicijalizirana u okolini poznatih područja izražene dobrote. Ugrađivanje znanja u postupak inicijalizacije početne populacije zahtijeva ulaganje određenog vremena, kako za istraživanje domene rješenja, tako i za samu implementaciju računalnog koda. U literaturi postoje kontradiktorni rezultati o isplativosti implementacije ovoga znanja, no, generalno, u slučaju optimizacije s ograničenjima, napor potreban za implementaciju isplativ je. Dodatni razlog zašto se u većini teoretskih razmatranja evolucijskih algoritama inicijalizacija izvodi po slučaju je taj što se ti algoritmi testiraju na tzv. *benchmark* problemima, za koje su unaprijed poznati položaji optimuma, te bi uključivanje ekspertnog znanja u bilo kojem dijelu pretrage prostora bilo zapravo varanje.

2.3 Genetski algoritmi

Genetski algoritmi (GA) najrašireniji su i najpoznatija klasa evolucijskih algoritama. Prvi put spominje ih John Holland, 1975. godine [35]. Također su i najistraživaniji iz skupine

kanonskih evolucijskih algoritama, pa su mnoge teoretske postavke, razvijene za njih, proširene i prilagođene ostalim klasama evolucijskih algoritama.

Kod GA često se koriste binarno kodiranje, što je posebno izraženo kod prvih njihovih primjena. Kromosomi su predstavljeni nizovima nula i jedinica, no danas se općenito prihvaća da je kodiranje kod genetskih algoritama temeljeno na bilo kojem konačnom alfabetu. Kodiranje, koje oslikava *genotipski* prostor uzrokuje potrebu za inverznom transformacijom – dekodiranjem, čime se dolazi u *fenotipski* prostor. U genotipskom prostoru izvode se svi evolucijski operatori osim selekcije. Selekcija odnosno vrednovanje pojedinaca iz populacije izvodi se u genotipskom prostoru, prostoru rješenja.

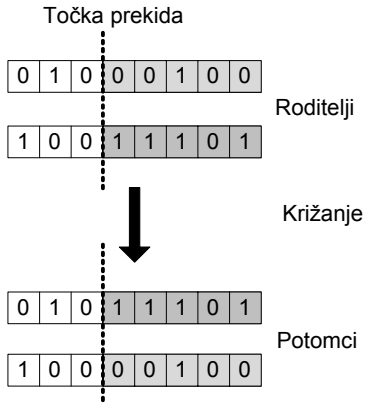
Uobičajeno kod GA potrebno je definirati funkciju dobrote, kojom je moguće razlikovati kvalitetu svakog člana populacije i izvršiti selekciju pojedinaca za roditeljsku populaciju na temelju vrijednosti dobrote. Standardna metoda selekcije je *selekcija proporcionalna dobroti* koja se često ostvaruje primjenom *ruletnog pravila.* Vjerojatnost izbora $P(i \mid f_i)$, pojedinca i, proporcionalna je vrijednosti njegove dobrote f_i, normaliziranoj prema ukupnoj dobroti populacije N pojedinaca:

$$P\left(i \mid f_i\right) = \frac{f_i}{\sum_{j=1}^{N} f_j}, f_i \geq 0. \tag{3}$$

Jedan od nedostataka ovog pristupa je da je moguće, kod nekih rješenja, koja u ranoj fazi evolucije postižu visoke vrijednosti dobrote, da prevladaju, odnosno budu ponavljano izabrana u sljedeću populaciju. Time se smanjuje opseg pretraživanja i zastaje se u podprostoru koji ne mora biti blizu globalno-optimalnog. Jedno od rješenja za uklanjanje ovog problema je primjenom metoda *skaliranja* dobrote, kojima se značajna razlika dobrote određenih pojedinaca može umanjiti. Drugi je problem kada različita rješenja postižu slične vrijednosti na skali dobrote. U tom slučaju izostaje evolucijski pritisak na populaciju i nema vođenja evolucije u smjeru željenog rješenja.

Sljedeća metoda koja se često koristi s ciljem selekcije, posebno kada je populacija potencijalnih rješenja velika, je *turnirska selekcija*. Kod turnirske selekcije, uz turnir veličine k, $k \geq 2$, k pojedinaca se po slučaju izabire iz populacije, pojedinac s najvećom vrijednosti dobrote izabire se za roditeljsku populaciju.

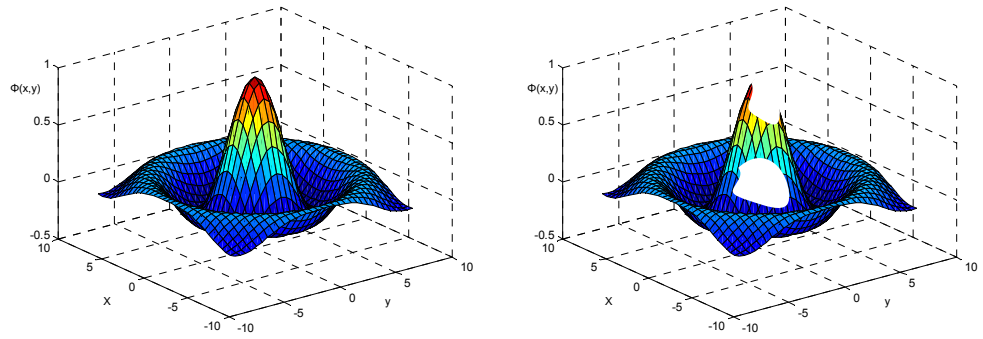
Nakon izbora dovoljnog broja pojedinaca, što je N uz pretpostavku da je populacija konstantne veličine i da se prethodna generacija u potpunosti mijenja novom, na novoj se populaciji primjenjuje operator rekombinacije.



Slika 2-7. Križanje u jednoj točki između dva kromosoma kodirana binarnim nizovima.

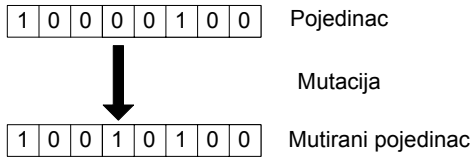
Rekombinacija se odnosi na izmjenu genetskog materijala između dva člana roditeljske populacije. Najjednostavniji oblik rekombinacije je *križanje u jednoj točki*, koje se izvodi na dva roditelja. Postupak je takav da se po slučaju izabere točka prekida, a zatim se zamijene odgovarajući dijelovi roditelja ispred i poslije točke prekida, tvoreći tako dva nova rješenja. Naziv *rekombinacija* podrazumijeva općenito veći broj članova roditeljske populacije na ulazu, npr. četiri, dok se križanje uvijek izvodi na dva člana.

Nakon križanja, primjenjuje se na potomke operator mutacije. Kod binarnog kodiranja, mutacija se izvodi tako da se s pred definiranom vjerojatnošću mijenja vrijednost svakog elementa niza koji kodira pojedinca i to iz 1 u 0 ili obratno. Uloga mutacije je izuzetno bitna za konvergenciju algoritma jer osigurava povezanost prostora koji se pretražuje. Primjenom samo operatora križanja, obzirom da je populacija po definiciji konačne veličine, broj rješenja koje se križanjem mogu dobiti opet je konačan broj. Kod kontinuiranih funkcija posljedica je da je broj lokacija na kojima se vrijednost funkcije može ispitati konačan broj, što zapravo znači da je dio informacija izgubljen odnosno da nije moguće doći u sve točke tog prostora.



Slika 2-8. Primjer povezanog (lijevo) i nepovezanog prostora pretrage (desno).

Postupak evaluacije, selekcije, križanja i mutacije ponavlja se dok se ne zadovolji uvjet prekida. Pri tome uvjet prekida može nastupiti kada se dosegne maksimalno dopušteni broj generacija algoritma, kada se pronađe optimalno rješenje ili kada primjerice standardna devijacija populacije kroz određeni broj uzastopnih generacija iznosi manje od neke prethodno određene vrijednosti.



Slika 2-9. Mutacija binarnog niza.

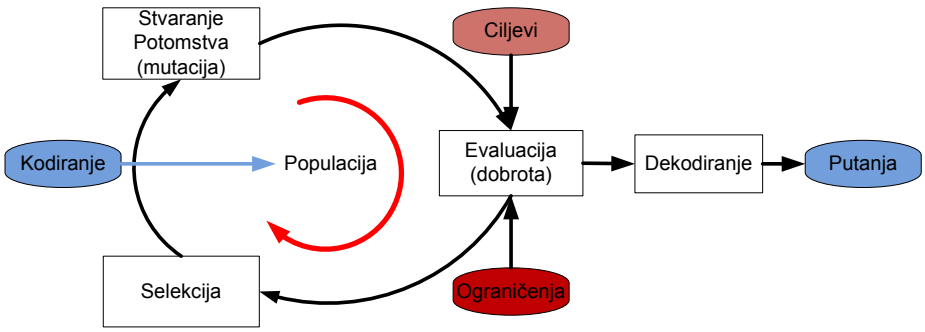
Naime, često se unaprijed ne zna vrijednost optimalnog rješenja nekog problema. Postavljanjem čvrste granice na broj dopuštenih generacija može se prekinuti algoritam u trenutku kada su se pojedinci populacije približili optimalnom rješenju, no ono još nije dosegnuto. Nasuprot tome, kada standardna devijacija populacije padne ispod određene vrijednosti, znači da je cijela populacija slična, odnosno nalazi se u okolini jednog optimuma, globalnog ili lokalnog, uz smanjenu mogućnost izlaska iz tog područja. Tada je utemeljeno prekinuti rad algoritma te ponovnim pokretanjem provjeriti o kakvoj vrsti optimuma se radilo u prethodnom koraku.

2.3.1 Jednostavan genetski algoritam za planiranje putanje mobilnog robota

Ilustracija rada genetskog algoritma pokazati će se na rješenju problema planiranja putanje za robota / agenta koji se nalazi u okolini s preprekama. Agent se može pomicati za unaprijed definirani iznos, korak, prema naprijed, natrag, lijevo i desno. Stoga se odabire kodiranje

sljedećeg alfabeta $\Omega = \{1,2,3,4\}$; pri čemu je smisao kodiranja sljedeći: 1 – inkrementalni pomak naprijed, 2 – inkrementalni pomak natrag, 3 – inkrementalni pomak lijevo i 4 – inkrementalni pomak desno.

Sljedeće je potrebno definirati funkciju dobrote. Želi se da robot najkraćim mogućim putem prijeđe put od početne do ciljne točke, i da pri tome ne kolidira s preprekama iz radnog prostora.



Slika 2-10. Rješavanje problema planiranja putanje jednostavnog agenta primjenom genetskog algoritma.

Izbor funkcije dobrote je, kao što je već rečeno slobodan, načelno je potrebno uzeti što manji skup parametara, koji je dovoljan za evoluiranje željenog rješenja. U prikazanom je primjeru odabrana funkcija sljedećeg oblika:

$$\Phi = w_1 \cdot C\left(\left(x,y\right)_c - \left(x,y\right)_g\right) + w_2 \cdot p, \tag{4}$$

pri čemu je C cityblock funkcija udaljenosti, w_1 i w_2 težinski faktori s vrijednošću iz intervala $\langle 0,1]$, $\left(x,y\right)_c$ je trenutna koordinata referentne točke robota, a $\left(x,y\right)_g$ je konačna koordinata referentne točke robota. Cityblock funkcija odabrana je iz razloga što prirodno odgovara metrici prostora, no moguće bi bilo odabrati i drugu mjeru udaljenosti, poput euklidske. Za kartezijski prostor tipa R^2 cityblock funkcija definirana je kao:

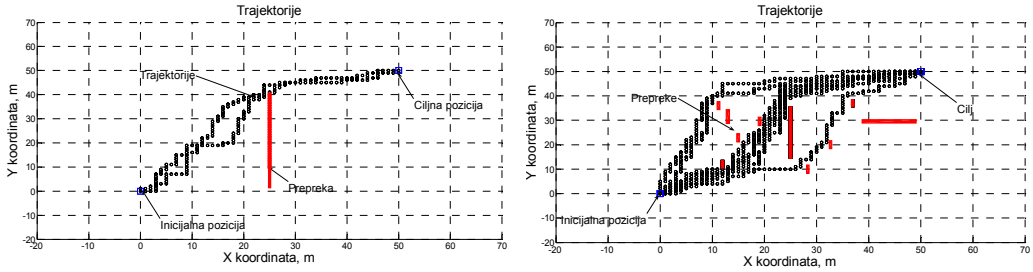
$$C = \left|x_1 - x_2\right| + \left|y_1 - y_2\right|. \tag{5}$$

Vrijednost p odnosi se na broj kolizija kojima određena trajektorija rezultira. Ovaj je algoritam relativno jednostavan za implementaciju i rezultati koji su njime dobiveni zadovoljavajući su u smislu kompletnosti rješenja [41]. Primijećeno je da učinak algoritma u velikoj mjeri ovisi o strukturi prepreka. Najteži problem predstavljale su točkaste prepreke stohastički raspoređene po prostoru, dok se u prostorima bez prepreka brzina konvergencije značajno povećavala.

Tablica 2-1. Učinak predloženog genetskog algoritma uz različite parametre.

p_c	p_m	N	$G_{srednje}$	σ
0,7	0,001	8	1547	642
0,7	0,001	16	1054	235
0,7	0,0005	32	1004	145

Tablica [Tablica 2-1] prikazuje parametre genetskog algoritma, s lijeva: vjerojatnost križanja, vjerojatnost mutacije, veličinu populacije i dobivene rezultate u vidu broja generacija potrebnih za pronalaženje rješenja i standardnu devijaciju potrebnog broja generacija dobivenih kroz seriju simulacija.



Slika 2-11. Putanje dobivene genetskim algoritmom.

Na slici [Slika 2-11] prikazane su dvije okoline i skupina rješenja u obliku trajektorija koje rješavaju pripadajuću okolinu u obliku putanje koja ne rezultira kolizijama. Više rješenja predstavljaju različite izlaze evolucijskog algoritma.

2.3.2 Teoretske osnove genetskih algoritama

Iako se *GA* koriste već više četiri desetljeća, i na razini implementacije su intuitivni i lako razumljivi, teorija koja se nalazi u njihovoj pozadini, kao i odgovori na pitanja kako oni u biti rade i zašto su djelotvorni, do danas je ostala na relativno skromnoj razini i nije se značajnije mijenjala od 1975. godine kada je John Holland [35] postavio *shema teorem*.

Shema predstavlja hiperravninu u prostoru koji se pretražuje i uobičajeno se za binarne alfabete označava simbolom #. Ovaj simbol označava da na mjestu u kromosomu, na kojem se simbol nalazi, može biti bilo koji element alfabeta. Za kromosom duljine l=5, shema 11### je hiperravnina koja ima jedinice na prva dva mjesta. Svi znakovni nizovi koji zadovoljavaju

ovu shemu nazivaju se njezine *instance* ili *primjeri*, a u promatranom slučaju, za binarni alfabet ima ih 2^3=8.

Dobrota sheme definirana je kao srednja vrijednost dobrote svih znakovnih nizova koji su njezini primjeri. U praksi, kada shema u zadanoj populaciji ima puno instanci, ova vrijednost može se procijeniti uzorkovanjem ograničenog podskupa skupa svih primjera. Globalna optimizacija može se sada prikazati kao i kao traženje sheme koja nema simbola #, kojoj su sve vrijednosti fiksirane i koja onda prema definiciji ima najveću dobrotu.

Rani radovi Johna Hollanda pokazali su da je analiza genetskih algoritama moguća primjenom koncepta sheme. On je pokazao da je niz znakova duljine l instanca 2^l shema. Iako u općem slučaju u populaciji veličine N neće biti prisutno $N \times 2^l$, pokazao je da će populacija uspješno procesirati $\Theta\left(N^3\right)$ shema. Ta spoznaja, koja se naziva *implicitnim paralelizmom* općenito se smatra jednim od glavnih razloga uspješnosti genetskih algoritama [42].

Shemu se opisuje pomoću dva obilježja, reda – broj mjesta u nizu znakova koji su fiksirani, odnosno, na njima nije znak #, i duljine, udaljenost između dva najudaljenija člana niza. Duljina d ujedno i određuje maksimalni mogući broj mjesta prekida nekog kromosoma. Shema H=1##454### je reda 4, $o\left(H\right)=4$, a duljine 9-1= 8, $d\left(H\right)=8$.

Broj instanci neke sheme u evoluirajućoj populaciji ovisan je o utjecaju varijacijskih operatora. Dok operator selekcije može utjecati jedino na učestalost pojavljivanja neke sheme u dolazećim populacijama, operatori križanja i mutacije mogu kreirati nove primjere ili uništiti postojeće. U tekstu koji slijedi, $P_d\left(H,x\right)$ označava vjerojatnost da operator x uništi shemu H, a $P_s(H)$ označava vjerojatnost izbora niza koji sadrži instancu sheme H.

2.3.3 Shema teorem za standardni genetski algoritam

Početne postavke: selekcijski mehanizam je temeljen na izboru proporcionalnom dobroti, križanje je u jednoj točki, u daljnjem tekstu 1X, mutacija je binarna s vjerojatnošću P_m, te je zamjena generacijska, odnosno nema čuvanja članova iz prethodne generacije. Ako se pretpostavi da kromosom duljine l sadrži primjer sheme H, shema može nestati uslijed križanja u jednoj točki, ako točka prekida padne unutar krajeva sheme sa sljedećom, vjerojatnošću:

$$P_d\left(H,1X\right)=\frac{d\left(H\right)}{l-1}. \tag{6}$$

Vjerojatnost da binarna mutacija, uništi shemu *H* proporcionalna je redu sheme *H:*

$$P_d\left(H,mutacija\right)=o\left(H\right)\cdot P_m. \tag{7}$$

Vjerojatnost da će shema *H* biti izabrana u roditeljsku populaciju ovisi o relativnom odnosu dobrote pojedinaca iz populacije koji sadrže shemu *H*, prema iznosu dobrote cijele populacije i o broju primjera sheme u trenutnoj populaciji $n\left(H,t\right)$. Ako se s $f\left(H\right)$ označi dobrota sheme *H*, koja je definirana kao srednja vrijednost dobrote svih pojedinaca koji su instance sheme *H*, a s $\tilde{f}$ srednja vrijednost dobrote cijele populacije, dobiva se:

$$P_s\left(H,t\right)=\frac{n\left(H,t\right)\cdot f\left(H\right)}{N\cdot\tilde{f}}, \tag{8}$$

što je vjerojatnost da će niz koji sadrži shemu *H* biti izabran u sljedeću populaciju, uz uvjet da se bira *N* neovisnih članova postojeće populacije te da se *N* ne mijenja. Ukupan očekivani broj instanci sheme *H* u populaciji nakon selekcije je sada:

$$\tilde{n}\left(H,t\right)=N\cdot P_s\left(H,t\right)=\frac{n\left(H,t\right)\cdot f\left(H\right)}{\tilde{f}}. \tag{9}$$

Nakon dijeljenja s parametrom veličine populacije, *N*, kako bi se rezultati učinili neovisnim o broju članova populacije, uzimajući u obzir moguće uzroke uništenja sheme putem križanja i mutacije, i uvodeći znak nejednakosti s ciljem dopuštanja stvaranja novih primjera sheme *H*, kao posljedicom istih varijacijskih operatora, udio $m\left(H\right)$ jedinki koji sadrže shemu *H* u budućim vremenskim trenucima određen je sljedećim izrazom:

$$m\left(H,t+1\right)\geq m\left(H,t\right)\cdot\frac{f\left(H\right)}{\tilde{f}}\cdot\left[1-\left(p_c\cdot\frac{d\left(H\right)}{l-1}\right)\right]\cdot\left[1-p_m\cdot o\left(H\right)\right], \tag{10}$$

pri čemu su p_m vjerojatnost mutacije, a p_c vjerojatnost primjene operatora križanja. Izraz prikazan jednadžbom (10) predstavlja shema teorem čije je tumačenje da će sheme koje postižu nadprosječnu dobrotu povećati broj svojih instanci tijekom umjetnog evolucijskog procesa.

2.4 Koevolucijski algoritmi

Evolucijski algoritmi su uspješno primijenjeni na različitim domenama problema, od čisto optimizacijskih primjena na pronalaženje optimuma niza funkcija, do stvarne primjene kod primjerice strojnoga učenja. Kako uspjesi tako i neuspjesi različitih primjena ove skupine algoritama doveli su do novih spoznaja pa i poboljšavanja evolucijskih algoritama. Prirodna i u novije vrijeme vrlo popularna metoda istraživanja koja se može primijeniti kada su problemski prostori vrlo kompleksni, ili kada je teško kreirati odgovarajuću funkciju cilja, je metoda koevolucijskih algoritama. Kod ove vrste algoritama, dobrota je mjera koja evaluira interakciju pojedinaca iz različitih populacija. Ova specifičnost daje mogućnost evoluiranja kompleksnijih agenata zbog toga što se dijelovi problema rješavaju u parovima koji su u međusobnoj interakciji. Isto tako, omogućuje se rješavanje problema gdje je ovakav, subjektivan način mjerenja dobrote nužan, npr. kod razvijanja strategija igara. Dobrota jednog pojedinca ovisna je o dobroti pojedinca s kojim se uspoređuje i promjena u distribuciji koevoluirajućih populacija utječe na promjenu svih pojedinaca te populacije.

2.4.1 Definicija koevolucije

Koevoluciju se može definirati prema [43] kao recipročno induciranu evolucijsku promjenu između dviju ili više vrsta ili populacija. Iako je ova definicija intuitivna kada se razmatra u svjetlu bioloških sustava, pojam koevolucije u području evolucijskih algoritama još je prilično nejasan i nedovoljno određen. Jedno od pitanja oko kojega je u znanstvenoistraživačkoj zajednici postignut konsenzus je to da je izračun dobrote kod koevolucijskih algoritama *subjektivan*. Kod standardnih evolucijskih algoritama, dobrota se mjeri *objektivno*, odgovarajućom funkcijom dobrote mjeri se svaki pojedinac iz populacije.

Kod koevolucijskih algoritama nameću se pitanja poput: kakve je prirode interakcija među pojedincima, moraju li pojedinci biti razmješteni u različite populacije ili biti članovi iste, mogu li pojedinci biti uzimani u trenutnu populaciju kao članovi neke bivše, već izumrle populacije su utemeljena i danas postoje u istraživačkoj zajednici prijepori u odgovorima na njih.

Prema navedenim razmatranjima, prirodna evolucija je uvijek koevolucijski proces jer je dobrota pojedinca uvijek vezana uz pojedince s kojima se nalazi u zajedničkom prostoru. Sljedeće četiri definicije uvode se kako bi se izbjegle neodređenosti u različitim načinima evaluacije pojedinaca:

Definicija 1. ***Objektivna mjera*****:** ***mjera*** *koja evaluira pojedinca* ***neovisno*** *od svih ostalih pojedinca, osim kod efekata skaliranja i normalizacije.*

Definicija 2. ***Subjektivna mjera*****:** ***mjera*** *koja nije objektivna.*

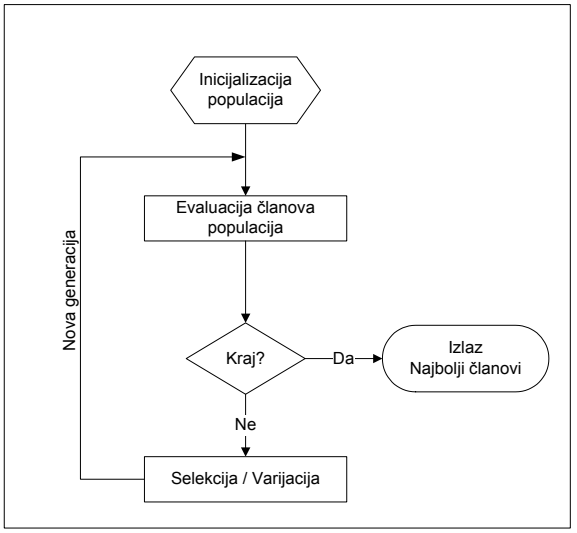
Definicija 3. ***Unutrašnja mjera:*** *mjera koja utječe na* ***smjer evolucije*** *na bilo koji način.*

Definicija 4. ***Vanjska mjera:*** *mjera koja ne može utjecati na* ***smjer evolucije*** *na bilo koji način.*

Sada je jasno da je dobrota kada se koristi u području istraživanja evolucijskih algoritama uvijek unutrašnja mjera. Vanjske mjere su vezane uz primjerice statističko praćenje srednje dobrote populacije, ili informacije vezane uz najbolje pronađeno rješenje do nekog trenutka.

Na temelju definicija 1 - 4 koevolucijski se algoritmi mogu definirati na sljedeći način:

Definicija 5. Koevolucijski algoritam – *je evolucijski algoritam koji za procjenu dobrote koristi subjektivnu unutrašnju mjeru.*



Slika 2-12. Dijagram toka osnovnog koevolucijskog algoritma.

2.4.2 Kooperativna i kompetitivna koevolucija

Obzirom da se koevolucijski algoritmi razlikuju od tradicionalnih evolucijskih algoritama po tome što se koristi subjektivna mjera dobrote pri čemu se pojedinci ocjenjuju na temelju interakcija s drugim pojedincima, postavlja se pitanje kakve vrste mogu biti ove interakcije. Postoje dvije osnovne vrste interakcije, *kooperativna* i *kompetitivna*.

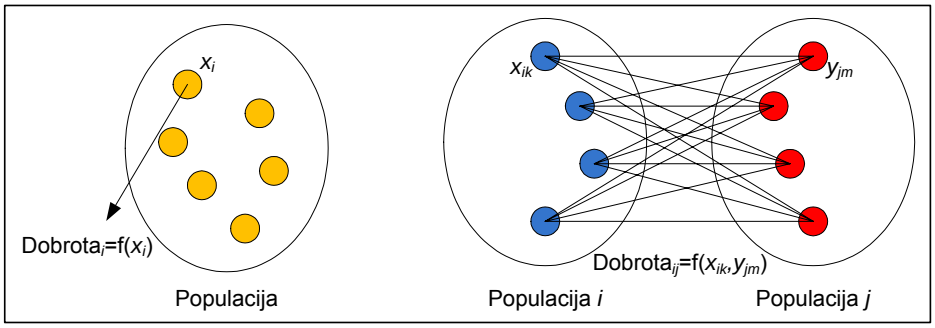
Može se zamisliti primjer gdje se pojedinci ili populacije natječu ili bore jedni protiv drugih. Tipičan je primjer model tipa *grabežljivac–plijen*. Ovaj model primijenjen na kontekst planiranja putanje mogao bi izgledati ovako: jedna populacija predstavlja instance upravljačkog programa robota. Druga populacija predstavlja okoline kroz koje se robot treba kretati. Zadatak je prve populacije da razvije što bolja rješenja za okoline iz druge populacije. Zadatak je druge populacije da razvije što kompleksnije okoline kroz koje se robot mora kretati. Ovakva vrsta interakcije naziva se kompetitivnom interakcijom. Rezultat je da razvojem što složenijih okolina, upravljački algoritam robota postaje sve bolji i sposoban rješavati složenije zadaće.

Sljedeći bi primjer mogao biti razvoj upravljačkih programa za agente koji su u potrazi za hranom. Svaki agent treba razviti strategiju prikupljanja što veće količine hrane, u uvjetima ograničenih resursa, natječući se s ostalim agentima.

Druga vrsta primjera, ona koja uključuje kooperativno djelovanje je ona gdje svaka populacija predstavlja dio rješenja nekog većeg problema i zadatak je populacija da razviju što bolje dijelove rješenja nekog većeg, holističkog problema. Ideja je da se kompleksniji, kompozitni problem podijeli, te da je svaka koevoluirajuća populacija zadužena za određeni dio problemskog prostora. Ako se ovaj problem ilustrira primjerice funkcijom od dvije promjenjive $\Phi = f(x, y)$, prva populacija može biti zadužena za razvijanje dijela x, dok je shodno tome druga populacija zadužena za razvijanje dijela rješenja y.

Važan doprinos razvoju kompetitivnih evolucijskih algoritama je rad [44] gdje je prikazan pristup koevolucije razvrstavajućih mreža i uzoraka temeljen na konceptu grabežljivac-plijen. Razvrstavajuće mreže razvijaju se uz drugu populaciju koja predstavlja skupove podataka koje je potrebno razvrstati. Pojedincu iz jedne populacije, koji predstavlja jednu moguću razvrstavajuću mrežu, dodjeljuje se dobrota prema tome kako je dobro razvrstao oponenta iz druge populacije. Pojedinci iz druge populacije dobivaju dobrotu vezano uz to koliko dobro onemogućuju svoje razvrstavanje oponentima iz prve populacije.

Danas je velika većina radova u području koevolucijskih algoritama usmjerena na istraživanje kompetitivne koevolucije i ti se radovi većinom bave problemom razvijanja strategija igara.



Slika 2-13. Vrednovanje pojedinca kod standardnog evolucijskog (lijevo) i koevolucijskog algoritma (desno).

Istraživanja u području kooperativne koevolucije započinju u prvo vrijeme primjenom na optimiranje statičkih funkcija, a potom i na učenje neuronskih mreža [45]. U ovom slučaju jedna populacija predstavlja potencijalne arhitekture neuronske mreže, dok druga populacija zadužena za razvijanje sinaptičkih veza. Arhitekture mreže evaluirane su prema tome koliko dobro rješavaju zadani problem sa sinapsama iz druge populacije i članovi druge populacije dobivali bi dio dobrote članova prve populacije. Stoga su sinapse više nagrađivane ako bi bile prisutne kod uspješnijih arhitektura mreže.

Razlike između kompetitivne i kooperativne koevolucije nisu zanemarive, i iako postoje mnoge sličnosti, ove dvije vrste algoritama sklone su različitim vrstama patologija, ali imaju i različite prednosti jedni pred drugima.

Kao što je ranije već objašnjeno, kod koevolucijskih algoritama dobrota svakog pojedinca iz prve populacije, u kanonskom slučaju evaluacije svakog člana prve populacije sa svakim članom druge populacije, ovisna je o strukturi druge populacije, kao što je prikazano na slici [Slika 2-13]. Kumulativna dobrota pojedinca *i* iz prve populacije računa se prema izrazu:

$$f_i = \sum_{j=1}^{N} G(i,j), \tag{11}$$

pri čemu se *j* odnosi na trenutnog člana iz druge populacije, *N* je ukupni broj članova iz druge populacije, a *G* označava interakciju između članova različitih populacija.

Obzirom da je kanonsko vrednovanje vrlo zahtjevno s aspekta računalnih resursa, iako jedino ono daje ukupnu informaciju o stanju u obje populacije, u uporabi se često nalazi i vrednovanje temeljeno na konceptu *svi sa prethodno najboljim(a)* i razne varijacije turnirskog vrednovanja.

2.4.3 Neke prednosti koevolucije

Ističu se dvije domene problema gdje bi koevolucijski algoritmi mogli pokazati bolji učinak od standardnih evolucijskih algoritama [46]:

- problemi s velikim, (beskonačnim) prostorima pretrage i
- problemi kod kojih je teško odrediti objektivnu mjeru dobrote.

U oba navedena slučaja, nada u rješavanje problema temelji se na očekivanom razvoju mehanizma *arms race*[iv] koja bi se trebala stvoriti među populacijama i osigurati dinamiku evolucijskog pritiska. Kod ovog se mehanizma odvija konstantan inkrementalni napredak svake populacije u odnosu na ostale, na takav način da je osiguran stalan napredak svih populacija. Sustav se uslijed toga kreće inkrementalno prema boljim područjima prostora rješenja. U jednostavnom scenariju vrste grabežljivac – plijen: ako plijen evoluira tako da postane brži, tada je populacija predatora prisiljena razviti takve vrste ponašanja koji anuliraju ovo obilježje plijena. Ako sada populacija predatora razvije bolji osjet njuha, kako bi prije uočila plijen, tada se opet plijen treba mijenjati. Očekivanje i nada je koevolucije da će na kraju procesa obje populacije imati svojstva vrlo visoke kvalitete.

Problemi s velikim (beskonačnim) prostorima pretrage

Kod tradicionalnih jednokriterijskih optimizacijskih problema, rješenje problema predstavljaju argumenti funkcije koji daju najveću vrijednost funkcije dobrote. Neki problemi zatvaraju posebno velike, čak i beskonačno velike domene prostora rješenja. U takvom slučaju, logično je očekivati da nema takve optimizacijske metode koja može u realnom vremenu pretražiti prostor rješenja i pronaći globalno optimalno područje. U takvom slučaju, od interesa je pronaći kvalitetne pod-prostore ukupnog prostora rješenja.

Primjer je rješenje problema razvrstavajućih mreža ilustrirano prethodnim poglavljem. Ukupni je prostor pretrage vrlo velik. Ako bi se standardnim evolucijskim algoritmom pretraživao prostor svih mreža koje ispravno sortiraju sve moguće skupove podataka, rezultat bi vjerojatno bio neuspješan. Jedno moguće rješenje tog problema je da se odabere stacionarni podskup određenih skupova podataka koji se koriste kao primjeri za učenje sortiranja. Problem je što će evolucijski algoritam biti prilagođen radu za odabrane, specifične skupove podataka, ali ne nužno i za neki novi, nepoznati skup. Drugi mogući pristup je da se uzorci skupova za učenje odabiru stohastički, po slučaju. Obzirom na veličinu prostora iz kojega je

[iv] Ovaj mehanizam (utrka u naoružanju) danas se većinom koristi pri opisu procesa nadmetanja bez stvarnog, objektivnog cilja, osim biti bolji od drugih natjecatelja.

moguće generirati skupove podataka, evolucijski algoritam u tome slučaju ne nauči zapravo ništa korisno.

Koevolucijski algoritmi imaju drugačije obilježje: to je metoda koja prilagođava oba dijela prostora rješenja zajedno i istovremeno jedno drugome. Očekuje se u primjeru razvrstavajućih mreža da će algoritam sam pronaći odgovarajuće pod-prostore prostora skupova podataka i da će ti referentni prostori biti odgovarajući primjeri koji će rezultirati kvalitetnim sortiranjem izvan ovih referentnih skupova.

Problemi bez karakteristične, objektivne mjere dobrote

Sljedeća skupina problema koju je vrlo teško riješiti primjenom standardnog evolucijskog algoritma je skupina problema koju je teško karakterizirati nekom objektivnom mjerom kvalitete. Ovo je često slučaj kod strategije igranja različitih igara. U mnogim slučajevima, igre pokazuju netranzitivno svojstvo, čime se vrednovanje igrača komplicira. Naime, neka strategija *A* može biti bolja od strategije *B*, strategija *B* može pobijediti strategiju *C*, dok strategija *C* može pobijediti strategiju *A*.

Ovo svojstvo netranzitivnosti može biti vezano ili uz dijelove prostora stanja ili može prevladavati i biti karakteristika čitavog prostora. U drugom slučaju, može se postaviti pitanje što je cilj strategije, razvijanje strategija koja pobjeđuje čim više protivnika, ili razvijanje strategije koja je u stanju pobijediti neke visoko kvalitetne protivnike. Za rješavanje ovog problema primjenom standardnih EA, moguć je pristup definiranjem određenog skupa strategija i učenjem na tom skupu, uz nadu generalizacije.

Međutim, u slučajevima kada je teško odrediti objektivnu mjeru dobrote, koevolucija ima na raspolaganju nešto više od ko-adaptacije, ima implicitan odgovor na pitanje „što je najbolje“. Naime, pokazano je da su koevolucijski algoritmi predisponirani za pronalaženje Nashovih ravnotežnih stanja, što im daje mogućnost pronalaska onih strategija koje su najmanje moguće dominirane[v] [47].

2.4.4 Patologije koevolucijskih algoritama

Iako je koncept evolucije motivirajući i obećavajući, primjene koevolucijskih algoritama (kako kompetitivne tako i kooperativne) često nisu uspješne, barem u onoj mjeri kojoj se očekuje, ili se problemi manifestiraju u vrlo teškom ugađanju parametara, koje može biti

[v] Pojam dominacije rješenja objašnjava se pod poglavljem 4.7. *Evolucijski algoritmi i višekriterijski optimizacijski problemi.*

vremenski zahtjevnije nego kod tradicionalnih EA. Uzroci tome dijelom leže u prirodi evaluacije, koja je u ovom slučaju *subjektivna*, a dijelom u samoj složenosti koevolucijske dinamike. Ove dvije poteškoće u kombinaciji imaju za posljedicu često neshvatljivo i neobjašnjivo ponašanje koevolucijskih algoritama. Neke od tradicionalnih patologija, iako u literaturi relativno šturo obrađene su sljedeće:

- gubitak gradijenta,
- cikličnost, i
- problemi fokusiranja.

Kod *gubitka gradijenta* slučaj je takav da jedna od populacija postaje previše dominantna u odnosu na ostale. U tom slučaju, ostale populacije nemaju na raspolaganju dovoljno informacija na temelju kojih bi se mogle poboljšavati, odnosno učiti (u prirodi, djeca uče jednostavnije zadatke prema težima, osiguravajući tako postupnost i primjenu novostečenih znanja u kasnijoj fazi). Sljedeći problem je problem *cikličnosti*, pri kojemu netranzitivnost u vrednovanju pojedinaca može prouzročiti situaciju kada se jedna populacija neznatno mijenja, kako bi ostvarila prednost nad drugom, nakon toga mijenjaju se i ostale populacije, što može uzrokovati opet vraćanje prve populacije u početno stanje. Priroda nekih igara je takva da rješenja nema, odnosno da su ciklične – poznati primjer je igra škare kamen papir. Kod *problema fokusiranja* radi se o situaciji u kojoj su rješenja dobivena koevolucijskim algoritmom nestabilna, odnosno krhka. Ovo je posljedica mogućnosti da se neka populacije pre-specijalizira za rješavanje jedne poddomene problemskog prostora, u svjetlu strategija igara, iskorištava se neka slaba osobina protivnika. Posljedica pre-specijalizacije je to da tako razvijena strategija ne postiže dobre rezultate protiv strategija koje se razlikuju od onih prema kojima se specijalizirala.

Ove probleme nije jednostavno detektirati i često je pri radu koevolucijskog algoritma nemoguće razlikovati o kojoj se patologiji radi. Naime, nepostojanje objektivne mjere dobrote u značajnoj mjeri otežava analizu progresa koevoluirajućih populacija, pri čemu se, iako naizgled ima promjene u distribuciji populacija može zapravo odvijati cikličnost ili slično. Metode za detektiranje ovih patologija i njihovog uklanjanja predmetom su većeg broja istraživanja danas.

2.5 Računska složenost evolucijskih algoritama

Analiza računske složenosti evolucijskih algoritama počela je pobuđivati interese tek sredinom devedesetih godina prošlog stoljeća [48]. Prvi rezultati odnosili su se na jednostavne algoritme klase (1+1)-EA[vi], na tzv. *toy* problemima[vii]. Ova početna istraživanja polučila su dublje razumijevanje uspješnosti EA na različitim prostorima stanja, kao i matematičkih metoda koje, proširivanjem i prilagodbom, postaju alat za analizu složenijih EA primijenjenih na realističnijim problemima. U posljednje vrijeme, pokazalo se mogućim analizirati (1+1)-EA na problemima kombinatorne optimizacije, kao i realističnije EA koji sadrže populaciju rješenja na strukturiranim *toy* problemima.

Već je napomenuto da postoji raskorak između broja primjena EA i popratne teorije koja objašnjava dinamiku njihovog rada. Devedesetih godina prošlog stoljeća, s pojavom teorije Markovljevih lanaca, pojavljuju se prvi rezultati koji se bave konvergencijom EA u odnosu na vrijeme potrebno za rješavanje optimizacijskih problema. Obzirom da su EA stohastičke prirode, očito je da deterministička definicija konvergencije nije prikladna za njihovo opisivanje. Iz tog razloga koriste se pojmovi stohastičke konvergencije [49]. U idealnom slučaju, EA bi trebao biti u stanju s vjerojatnošću 1 pronaći rješenje zadanog problema u konačnom broju koraka, bez obzira na način inicijalizacije početne populacije. U tom slučaju, kaže se da algoritam dolazi globalnog optimuma u konačnom broju koraka. Ukoliko algoritam zadržava u populaciji to rješenje $\forall\, t_i > t_{i-1}$, kaže se da algoritam konvergira u optimum. Primjenom Markovljevih lanaca, pokazano je da jednostavni GA, koji primjenjuje mutaciju, križanje i selekciju proporcionalnu dobroti ne konvergira u globalni optimum, dok isti taj algoritam s uključenim *elitizmom* (bezuvjetnim prenošenjem najboljeg pojedinca iz prethodne generacije u sljedeću) konvergira [50]. Kasnije je ovaj rad proširen na način da su definirani opći uvjeti EA, koji, ako su zadovoljeni, garantiraju konvergenciju [51]. Rad je motiviran činjenicom da „nije potrebno odrediti kvantitativno egzaktan Markovljev model za svaku varijantu evolucijskog algoritma kako bi se istražila njegova granična ponašanja“. Dani su također i uvjeti za ne-konvergenciju.

Unatoč tome, za uvjete kada EA konvergira, analiza vremenskog omeđenja ne daje precizne informacije o očekivanom vremenu pronalaženja rješenja, odgovor je na razini

[vi] (1+1)-EA je najjednostavniji evolucijski algoritam. Sadrži samo jedan kromosom u roditeljskoj populaciji, od kojega se stvara jedan potomak, primjenom mutacijskog operatora te se u suštini radi o *evolucijskoj strategiji*.

[vii] Toy problem u matematici i informacijskim znanostima je vrsta problema koji nije od primarnog znanstvenog interesa, ali se koristi s ciljem razumijevanja klase složenijih problema.

generalne omeđenosti potrebnog vremena. U [52] provedena je analiza broja generacija koji je potreban da bi konvergencija bila garantirana s fiksnim stupnjem pouzdanosti (s određenom vjerojatnošću d), neovisno o samoj optimizacijskoj funkciji. Taj rad i radovi koji slijede pokazali su da je najbolja gornja granica omeđenosti koja se pouzdano može garantirati ista onoj koju postiže metoda pretrage po slučaju, odnosno isto je slučaju kada se generacije inicijaliziraju po slučaju u svakom koraku algoritma [53]. Neuspjeh ovih radova da pronađu odgovarajuća omeđenja potvrdile su mišljenja da u slučaju analize vremenske složenosti općih, o domeni neovisnih, heurističkih metoda pretrage, ipak u razmatranje treba uzeti i samu funkciju koja se optimira.

Sljedeća potvrda toj tezi pojavila se kada je u [54] dokazano da postoje funkcije za koje osnovni (1+1)-EA pronalazi globalni optimum u vremenu $\Theta\left(n^n\right)$ [viii]. Veza između same optimizacijske funkcije i vremena potrebnog za konvergenciju danas je općepoznata, te je pokazano da za linearne funkcije, za (1+1) EA vrijeme iznosi $\Theta\left(n \log n\right)$, također je pokazano da postoje unimodalni prostori pretrage za koje potrebno vrijeme eksponencijalno raste, čime se vjerovanje da su EA efikasni na svim unimodalnim problemima pokazalo netočnim [55].

Područje istraživanja računalne složenosti EA danas je aktivno istraživačko polje s mnogim neodgovorenim pitanjima, u prvom redu o vremenskom učinku EA na stvarnim, *NP* teškim optimizacijskim problemima. Istraživanja su usmjerena u dva smjera, analizi osnovnog (1+1)-EA na teškim problemima, te analizi realističnijeg EA, s pouplacijom rješenja, na jednostavnijim problemima. Također je neodgovoreno pitanje može li EA nadmašiti učinak specifične optimizacijske metode kreirane za rješavanje određene klase problema.

2.6 Evolucijska robotika

Evolucijski algoritmi inspirirali su istraživače u području robotike za rješavanje različitih problema. Danas postoji područje evolucijske robotike koje je karakterizirano primjenom evolucijskih algoritama s ciljem oblikovanja autonomnih robota, pri čemu se oblikovanje odnosi na upravljački robotski sustav, tijelo robota ili oboje [56], [57].

[viii] Ovdje se *n* odnosi na duljinu gena, oznaka *l* u ostalim poglavljima. Oznaka *n* je izabrana kako bi se zadržala konzistencija s označavanjem u teoriji računalne složenosti.

Motivacija za primjenu evolucijskih algoritama u području robotike dolazi iz objektivne složenosti razvoja adekvatnih upravljačkih algoritama, o čemu je već ranije bilo riječi. Obzirom da je za pronalaženje rješenja primjenom evolucijskih algoritama često potrebno značajno vrijeme u kojem je potrebno izvršiti velik broj evaluacija, evolucijska se robotika u pravilu temelji na simulaciji odnosno na traženju rješenja u off-line načinu rada.

Razlika između evolucijskog računarstva i evolucijske robotike je u tome što je kod evolucijske robotike teško unaprijed definirati funkciju dobrote. Naime ponašanje živih bića, pa i autonomnih robota je *emergentno* svojstvo, svojstvo koje nastaje iz velikog broja interakcija agenta s okolinom. Kod evolucijske robotike nije prisutno sveobuhvatno znanje o problemskom prostoru, već možda izolirane karakteristike. U [57] predlaže se definicija trodimenzijskog prostora pomoću kojega je moguće klasificirati funkcije dobrote za različite evolucijske kriterije:

- dimenzija *funkcionalno-bihevioralno* prikazuje mjeru u kojoj će funkcija dobrote imati na izlazu novorazvijeno ponašanje ili biti usmjerena na optimizaciju parametara nekog specifičnog upravljačkog algoritma,
- dimenzija *eksplicitno-implicitno* razlikuje broj varijabli koje su sadržane u funkciji dobrote. Eksplicitna funkcija sadrži više komponenti, dok implicitna sadrži manji broj komponenti, i
- dimenzija *vanjsko-unutrašnje* ukazuje na to hoće li se agent oslanjati na globalne ili lokalne varijable stanja. Vanjska funkcija cilja uključuje globalne varijable koje su nedostupne agentu pri njegovom radu. Čisto unutrašnja funkcija cilja uključuje samo one podatke koje agent u svakom trenutku svoga rada može prikupiti slikom sa svojih senzorskih ulaza.

Funkcije dobrote standardne primjene evolucijskih algoritama smještene su u području funkcionalno-eksplicitno-vanjsko ovako definiranog 3D prostora, dok su u idealnom slučaju, za područje evolucijske robotike smještene u području bihevioralno-implicitno-unutrašnje, pri čemu je o domeni problema malo poznato.

Evolucijska robotika usporediva je s metodama strojnog učenja, npr. *metodom učenja pojačavanjem* (eng. *reinforcement learning*). Obje su metode upotrebljive za rješavanje iste vrste problema, ali razlikuju se prvenstveno u načinu traženja rješenja. Obje metode pretražuju prostor mogućih rješenja problema – načina djelovanja u određenim uvjetima, pri čemu sâmo rješenje problema može biti predstavljeno na različite načine, stablom odluke,

umjetnom neuronskom mrežom itd. Metode se prvenstveno razlikuju u tome što se način djelovanja svih pojedinaca mijenja po završetku zadatka u slučaju evolucijskog pristupa, dok se u slučaju učenja pojačavanjem agent adaptira kontinuirano tijekom izvođenja zadatka. Ova situacija odgovara i prirodnoj razlici između evolucije i učenja. Učenje predstavlja cjeloživotnu prilagodbu fenotipa, koji direktno ne utječe na promjenu genotipa, dok je evolucijom osigurana promjena i prilagodba genotipskih obilježja jedinki populacije tijekom generacija.

Kod evolucijske robotike ne pretpostavlja se neka određena vrsta upravljačkog procesa. U pravilu, bilo koji upravljački proces koji ima ugrađenu sposobnost evolvabilnosti – sposobnost slučajnih promjena s ciljem očekivanog poboljšanja [58], može se koristiti u evolucijskoj robotici. Tri osnovna pristupa u evolucijskoj robotici su:

- evolucija upravljačke umjetne neuronske mreže,
- optimizacija parametara već definiranog upravljačkog sustava, i
- evolucija samih računalnih programa primjenjujući genetsko programiranje.

Najčešći od navedenih je prvi pristup pri kojem se primjenom evolucijskog algoritma razvija određeni tip neuronskog upravljačkog mehanizma. Evolucija upravljačkih sustava temeljenih na umjetnim neuronskim mrežama opisana je u [59]. Zadatak koji je riješen je evolucija upravljačkog programa sposobnog za vođenje robota u labirintu. Cilj je robota bio kretati se čim brže te pritom izbjegavati sudare sa zidom labirinta. Funkcija dobrote oblikovana je tako da se nagrađuju pojedinci koji se kreću brzo i imaju minimalni broj sudara sa zidovima u jedinici vremena:

$$\Phi = V\left(1-\sqrt{\Delta v}\right)\cdot\left(1-i\right) \;\mid\; \left\{V,\Delta v,i\right\}\in\left[0,1\right], \tag{12}$$

pri čemu je V suma brzina oba kotača, Δv je apsolutna vrijednost razlike brzine dva kotača, i je normalizirana vrijednost očitanja infracrvenog senzora udaljenosti koji u danom trenutku ima najizraženiju aktivnost.

U prvim provedenim eksperimentima, korišten je robot Khepera, uz kodiranje vektorima realnih vrijednosti Elmanove upravljačke neuronske mreže [60]. Neuronska mreža sadržavala je 8 neurona na ulaznom sloju, koji su reprezentirali 8 Kheperinih infracrvenih senzora udaljenosti, te dva neurona sigmoidnog karaktera u izlaznom sloju, koji su reprezentirali brzinu vrtnje dva Kheperina kotača. Nakon pedesete generacije pronađeni su

pojedinci koji su demonstrirali zadovoljavajuće ponašanje koje je rezultiralo kontinuiranim kretanjem, bez kolizija sa zidovima labirinta.

Proširenje ovog eksperimenta uslijedilo je kada se neuronski upravljački mehanizam razvijen za Kheperu prenese na znatno većeg robota tipa Koala. Zadatak je istovjetan, potrebno je razviti kontinuirano kretanje po labirintu. Labirint je bio uvećan, ali i robot Koala imao je različiti raspored senzora po tijelu. Htjelo se ispitati adaptaciju neuro-upravljačkog mehanizma na novi senzorsko-motorni sustav. Upravljačka se mreža prvih 105 generacija evoluirala na Kheperi, a nakon toga prenesena je na Koalu, gdje je evoluirala dodatne 44 generacije. Nakon početnog naglog pada dobrote, bilo je potrebno 30 generacija za evolucijsku adaptaciju koja je rezultirala pojedincima sa sposobnostima sličnima onim najboljim pojedincima s Khepere nakon 105 generacija.

I u trećem eksperimentu iz ove serije, pri kojem je ponovno korišten robot Khepera, zadatak je bio pronaći takvu upravljačku neuronsku mrežu koja rezultira kontinuiranim kretanjem kroz labirint, no sada je fokus bio stavljen na istraživanje evolucije strojnog učenja. Umjesto evoluiranja sinaptičkih veza, kao u prva dva eksperimenta, evoluirala su se pravila učenja neuronske mreže. Svaka sinaptička težina adaptirala se tijekom cijelog procesa prema jednom od četiri pravila Hebbovog učenja:

$$w_t = w_{t-1} + \eta \cdot \Delta w_t, \tag{13}$$

gdje je parametar η genetski određen korak učenja sinapse, a promjena težine Δw_t računala se kao funkcija aktivacije predsinaptičkih i postsinaptičkih članova, prema izabranom Hebbovom pravilu. Rezultirajuća ponašanja i dobivene vrijednosti funkcije dobrote bile su slične u već opisanim eksperimentima bez učenja.

Primjer drugog pristupa evolucijskoj robotici, pri čemu je potrebno optimirati parametre definiranog upravljačkog programa je eksperiment prikazan u [61]. Evoluirana su dva načina dinamičkog kretanja robota AIBO, koji se kreće pomoću četiri noge. Predefinirani upravljački program kretanja sadržavao je 20 parametara realnih vrijednosti kojima se određivala pozicija i orijentacija tijela robota, smjer zanošenja robota, brzina nogu, disparitet između pozicije i orijentacije tijela robota, te tri dodatna parametra vezana uz način kretanja robota. Umjetna evolucija korištena je s ciljem optimizacije parametara kretanja stvarnog robota, nagrađujući brzo, ravnomjerno i pravocrtno kretanje. Dobrota pojedinca računata je kao produkt prosječne brzine i pravocrtnosti kretanja robota. Rezultati su eksperimenta bili raznoliki. Jedno od razvijenih kretanja nije bilo u punom smislu dinamičko, naime, robot je

vukao jednu prednju nogu po podu te nije mogao postići značajnu brzinu kretanja. U drugom slučaju, rezultat je bio zadovoljavajući, razvijajući kretanje dvostruko brže od dotada najbolje razvijenog kretanja dobivenog standardnim načinom.

Nekoliko studija bavi se primjenom genetskog programiranja kako bi se razvili sustavi upravljanja realnim robotima. Jedan primjer je [62], gdje je primijenjen poseban oblik genetskog programiranja kako bi se razvilo ponašanje izbjegavanja prepreka na Khepera robotu, koristeći slične postavke onima iz rada [59]. Predložena metoda genetskog programiranja djeluje izravno na strojnom kodu računala ugrađenog u robota. Razvijen je poseban operator križanja koji se može primijeniti na kromosome različite duljine koji sadrže strojne naredbe. Primijenjena je metoda stabilnog stanja, u kojoj se samo nekoliko pojedinaca iz populacije testira u svakoj iteraciji. Rezultati su eksperimenta impozantni. Potrebno je samo 40-60 minuta realnog vremena kako bi se zadovoljavajući program izbjegavanja prepreka razvio. Sličan pristup primijenjen je i na evoluciju upravljačkog algoritma implementiranu na vođenje letećih robota [63].

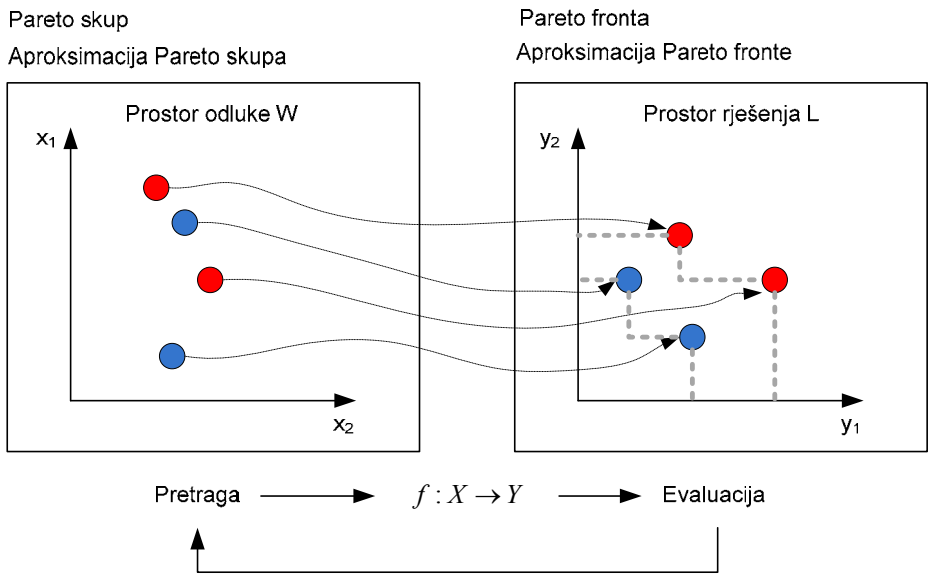
U [64] uveden je pojam sadržane evolucije u područje evolucijske robotike. Sadržana evolucija inspirirana je situacijom gdje se veći broj robota nalazi u međusobno slobodnoj interakciji za vrijeme dok izvode određeni zadatak u okolini koju dijele. Roboti generiraju potomstvo pri čemu je vjerojatnost da određeni roboti generiraju potomstvo vezana uz njihov uspjeh na zadatku. Sadržana evolucija oponaša distribuirana, asinkrona i autonomna svojstva prirodne evolucije. Osnovna razlika između sadržane evolucije i ostalih pristupa u evolucijskoj robotici je u tome što sadržana evolucija integrira reprodukciju s ostalim autonomnim ponašanjima. Zapravo, ovaj je pristup sličan koevolucijskom pristupu i od njega se razlikuje, samo po tome što se križanje izvodi među robotima čija je međusobna udaljenost manja od neke unaprijed određene granice. Pristup je ispitan na problemu foto taksije. Upotrijebljeno je 8 malih mobilnih robota kako bi razvili veoma jednostavnu upravljačku umjetnu neuronsku mrežu. Mreža je imala 2 neurona u ulaznom sloju, jedan binarni neuron koji je utvrđivao koji od dva osjetila robota prima jaču dozu svjetla, a drugi je bio bias. Ulazni su neuroni bili direktno spojeni na izlazne neurone kojima je bilo moguće putem četiri cjelobrojne vrijednosti upravljati brzinom vrtnje kotača. U ovom eksperimentu križanje među robotima nije bilo nešto čime se moglo direktno upravljati. Agent bi poslao poruku sa svojim genotipom, a drugi agenti koji su bili u dometu ove poruke mogli su primiti taj genotip. Selekcija je izvedena tako da su bolji agenti češće odašiljali broadcast poruke sa svojim genotipom i imali manje šanse za primanje takvih poruka od onih manje dobrih agenata.

U ovoj disertaciji zadatak je istražiti mogućnost primjene paralelnih, dijelom neovisnih algoritama na planiranje putanja dva robota u zajedničkom radnom prostoru. Za razliku od predloženih radova, koji se većinom bave proučavanjem ponašanja jednog robota, usmjerenim na razvoj upravljačke neuronske mreže, s ciljem razvoja općenitog ponašanja visoke razine u predloženoj se disertaciji traži strategije za dva robota – agenta koje osiguravaju vrlo preciznu koordinaciju među agentima istodobno optimirajući veći broj suprotstavljenih kriterija. Pitanja poput ograničavanja križanja među pojedincima ovisno o njihovom fizičkom rasporedu u radnom prostoru nisu predmetom razmatranja ove disertacije. Izloženi se problem formulira kao višekriterijski optimizacijski problem i traži se njegovo rješavanje sa što većom pouzdanošću.

2.7 Evolucijski algoritmi i višekriterijski optimizacijski problemi

Rješavanje višekriterijskih optimizacijskih problema (*VOP*) težak je zadatak. Kod ovih problema optimizacijski kriteriji često su suprotstavljeni i postojanje globalnog optimuma nije garantirano. Često su potrebni vrlo značajni računalni resursi kako bi se pronašlo zadovoljavajuće rješenje takvih problema. Poznate metode za rješavanje višekriterijskih problema optimiranja temelje se na metodama linearne agregacije funkcija i metoda temeljenih na Pareto pristupu. U pokušaju da se stohastički riješe problemi iz ove skupine u prihvatljivom vremenu, krajem osamdesetih godina prošloga stoljeća počinje se eksperimentirati s primjenom evolucijskih algoritama. Nakon ohrabrujućih rezultata, u literaturi se primjećuje značajno povećanje razvijenih aplikacija koje se temelje na evolucijskim algoritmima te se razvija i nova klasa evolucijskih algoritama usmjerenih na rješavanje višekriterijskih optimizacijskih problema, *MOEA*[ix] [65], [66].

[ix] Eng. Multi Objective Evolutionary Algorithm, MOEA, hrv. višekriterijski evolucijski algoritmi.



Slika 2-14. Prikaz općenitog višekriterijskog optimizacijskog problema.

Optimizacijski problemi kod kojih je prisutan jedan kriterij po kojem se vrednuje neko rješenje, mogu imati jedno, globalno-optimalno rješenje. Višekriterijski problemi u pravilu nemaju jedinstveno rješenje već *skup* rješenja koja nakon evaluacije predstavljaju vektore čije su komponente kompromisi u zadovoljavanju skupa kriterija u prostoru rješenja. Višekriterijski optimizacijski problem se bez gubitka općenitosti može opisati kao problem optimizacije s *k* kriterija, pri čemu su svi kriteriji jednako važni i sve je kriterije potrebno maskimizirati. Pretpostavlja se da se rješenje tako formuliranog problema može opisati pomoću *vektora odluke* $(x_1, x_2, ..., x_n)$, u *prostoru odluke* Ω. Funkcija $f : X \to Y$ vrednuje određeno rješenje dodjeljujući mu *vektor vrijednosti* $(y_1, y_2, ..., y_k)$ u *prostoru rješenja* L, kako je prikazano slikom [Slika 2-14].

Iako se koncepti Pareto optimalnosti često koriste, često su njihova tumačenja u literaturi nekonzistentna, radi konzistentnosti, definirat će se koncepte Pareto dominacije, Pareto optimalnosti i Pareto fronte kako slijedi:

Definicija 6. ***Pareto dominacija:*** *za vektor* $\vec{u} = (u_1, ..., u_k)$ *kaže se da dominira vektor* $\vec{v} = (v_1, ..., v_k)$*, što se označava kao* $\vec{u} \succeq \vec{v}$ *ako i samo ako je* $\vec{u}$ *djelomično veći od* $\vec{v}$*, odnosno* $\forall i \in \{1, ..., k\},\ u_i \geq v_i \wedge \exists i \in \{1, ..., k\} : u_i > v_i$.

Definicija 7. ***Pareto optimalnost:*** *Rješenje* $x \in \Omega$ *je Pareto optimalno s obzirom na* Ω *ako i samo ako ne postoji* $x' \in \Omega$ *za koje vrijedi da* $\vec{v} = f(x') = (f_1(x'), ..., f_k(x'))$ *dominira* $\vec{u} = f(x) = (f_1(x), ..., f_k(x))$.

Definicija 8. ***Pareto optimalan skup:*** *Za zadani VOP s* $f(x)$ *Pareto optimalan skup* $\mathcal{P}^*$ *je definiran sljedećim izrazom:*

$$\mathcal{P}^* := \left\{ x \in \Omega \mid \neg \exists\, x' \in \Omega : f(x') \succeq f(x) \right\}. \tag{14}$$

Definicija 9. ***Pareto fronta:*** *Za dani VOP s* $f(x)$ *i Pareto optimalan skup* $\mathcal{P}^*$*, Pareto fronta* $\mathcal{PF}^*$ *definirana je sljedećim izrazom:*

$$\mathcal{PF}^* := \left\{ \vec{u} = f(x) = \left(f_1(x), ..., f_k(x) \right) \mid x \in \mathcal{P}^* \right\}. \tag{15}$$

Pareto optimalna rješenja nazivaju se u literaturi još i neinferiornim ili efikasnim rješenjima; odgovarajući vektori nazivaju se nedominiranim vektorima. Ta rješenja ne moraju imati jasno izraženu povezanost, osim činjenice da pripadaju Pareto optimalnom skupu. Ova rješenja tvore skup svih rješenja čiji odgovarajući vektori nisu dominirani u odnosu na sve ostale poredbene vektore. Prikaz ovih vektora u prostoru rješenja predstavlja Pareto frontu. Pareto optimalan skup je podskup skupa svih mogućih rješenja u `W`. Preslikavanjem toga skupa u prostor rješenja `L` dobiva se Pareto fronta.

Pronalaženje stvarne Pareto fronte teško je i često se dobivena rješenja nalaze ili u blizini ili na rubnim dijelovima stvarne Pareto fronte. Kaže se da ta rješenja aproksimiraju Pareto frontu.

Kod EA razvijene su različite metode kojima se pomaže osiguravanju čuvanja rješenja koja dobro aproksimiraju Pareto frontu u uzastopnim generacijama. O ovome će biti riječi kasnije u disertaciji, prilikom implementacije evolutivnog algoritma.

U svjetlu problema razmatranog u ovoj disertaciji, razmatrani problem je višekriterijski, potrebno je istovremeno zadovoljavati četiri kriterija, od kojih su neki jasno suprotstavljeni (npr. kriteriji izbjegavanja kolizija i najkraće prijeđene udaljenosti referentne točke na izvršnom članu robota), dok se za ostale kriterije veza ne može sa sigurnošću unaprijed karakterizirati. Pareto frontu nije moguće predočiti grafički, jer se njezino rješenje nalazi u prostoru s pet dimenzija. Stoga se prikazuju projekcije Pareto fronte na odgovarajući broj osi kako bi se posredno predočio njezin oblik, kako je prikazano u poglavlju 3.2.8.

3 Razvoj višeagentskog modela upravljanja dvoručnim robotom

Model koji se u disertaciji razmatra odnosi se na dva robota koji dijele radni prostor i čije je gibanje potrebno koordinirati primjenom predloženog koevolucijskog algoritma. Agenti su definirani kao ravninski 2*R* (2 rotacijska zgloba) roboti inspirirani konstrukcijom SCARA konstrukcije. Svaki agent sastoji se od dva segmenta, pri čemu postoje rotacijski zglobovi između donjeg segmenta svakog robota s nepomičnom podlogom, te postoji rotacijska veza između donjeg i gornje segmenta robota. Krajnja točka gornjeg segmenta je slobodna i dalje će se označavati kao referentna, pretpostavljeno je da se na referentnoj točki nalazi tzv. *Tool Center Point*, što je u robotici uvriježen naziv za referentnu točku alata kojega robot u danom trenutku koristi.

U poglavlju 3.2 uvedena su određena geometrijska ograničenja, koja su proizašla iz formalnog razmatranja modela, a osiguravaju da:

- se sustav može smatrati višeagentskim te
- da se niz tranzicija svakog agenta može preslikati na realne robote.

Roboti su konačnih dimenzija te je u prvom redu potrebno osigurati izbjegavanje kolizija između segmenata lijevog i desnog robota. Ovaj je kriterij preduvjet uspješnog planiranja i ukoliko on nije ispunjen, putanje ne smiju biti prihvaćene kao rješenje problema. Obzirom da je problem po prirodi redundantan, odnosno da se ista točka prostora referentnom točkom može dosegnuti s više konfiguracija robota, uvode se kriterij minimizacije kuta zakreta u zglobovima robota. Pod pojmom *konfiguracija* podrazumijeva se skup podataka kojima je u određenom trenutku moguće jednoznačno odrediti položaj robota. Predloženi

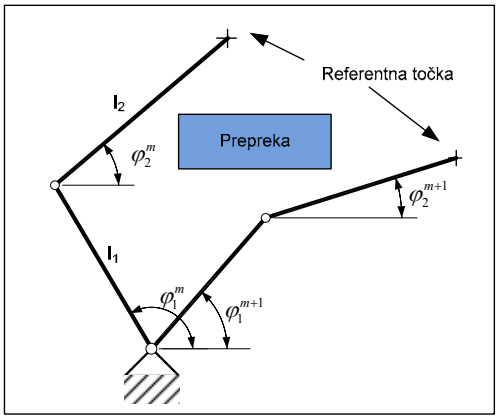
sustav inspiriran je stvarnim robotima, na kojima se želi i primijeniti, a kojima su iznosi najvećeg ubrzanja i usporenja omeđeni konačnim vrijednostima. Stoga je potrebno osigurati da izvršni član u određenom vremenskom intervalu ne prelazi udaljenosti veće od dopuštene, odnosno, traži se takav niz tranzicija izvršnog člana u prostoru, koji osigurava što ravnomjernije gibanje od poznate početne do poznate konačne konfiguracije.

Zadnji kriterij koji je ugrađen u model odnosi se na ukupno prijeđeni put što ga u ravnini prelazi referentna točka. Ovaj je kriterij u jasnoj vezi s kriterijem izbjegavanja kolizija, a veza je suprotstavljena. Naime, da bi putanje koje agenti u prostoru opisuju rezultirale kretanjem bez kolizija, u određenim se konfiguracijama agenti moraju jedan od drugoga odmaknuti. To rezultira većim utroškom vremena, koje je kod montažnih sustava često kritičan parametar, uključen putem vrijednosti *takt* koja opisuje najveće dopušteno vrijeme za sklapanje jednog proizvoda. Također, odmicanje utječe i na utrošak energije te na trošenje dijelova robota. Stoga je potrebno naći kompromis, koji je takav da agenti međusobno prilagode kretanje na takav način da niti u jednom trenutku ne dolazi do kolizija, a da je istovremeno vrijeme utrošeno na odmicanje i prilagodbu najkraće moguće.

Ukoliko se predoči SCARA konfiguracija robota, jasno je da kada se putanje oba agenta prilagode tako da u horizontalnoj ravnini nema kolizija, tada niti u vertikalnoj ravnini između dva robota ne može doći do kolizija, čime je zadatak planiranja gibanja agenata u radnom prostoru riješen.

Obzirom na složenost izloženog problema, njegova je implementacija sekvencijalna. U prvoj fazi predlaže se evolucijski algoritam te se s ciljem razumijevanja problema algoritam implementira na slučaju planiranja putanje jednog agenta u radnom prostoru. Optimizacijski kriteriji uvode se u koracima. U prvoj se fazi razmatra samo kriterij minimizacije prijeđenog puta. Rezultati dobiveni simulacijama jasno upućuju na potrebu uvođenja dodatnih kriterija optimizacije. Istodobno su uvođenjem dodatnih kriterija, ispituju se i razne postavke evolucijskih algoritama, koje je potrebno prilagoditi specifičnostima razmatranog problema. Ovo se u prvom redu odnosi na iscrpnu evaluaciju različitih oblika funkcije dobrote i pripadajućih težinskih parametara kojima je moguće dati naglasak određenim kriterijima. Također, s ciljem očuvanja aproksimacije, pronađene Pareto fronte, uvodi se lista u kojoj se čuvaju parcijalna najbolja rješenja mjerena svakim od četiri uvedena kriterija. Ova se rješenja kopiraju u roditeljsku populaciju ukoliko u trenutnoj generaciji nema boljih potencijalnih rješenja mjereno odgovarajućim kriterijem. Razmatraju se i ugađaju i ostali parametri čije

međudjelovanje nije intuitivno, poput veličine populacije, vjerojatnosti križanja, vjerojatnosti mutacije, načina selekcije.



Slika 3-1. Jedan agent u radnom prostoru uz prisutnu prepreku.

Nakon razumijevanja ovog algoritma prelazi se na modeliranje distribuiranog evolucijskog algoritma, čija je dinamika, kako je u prethodnim razmatranjima objašnjeno, značajno složenija.

3.1 Jedan agent u radnom prostoru

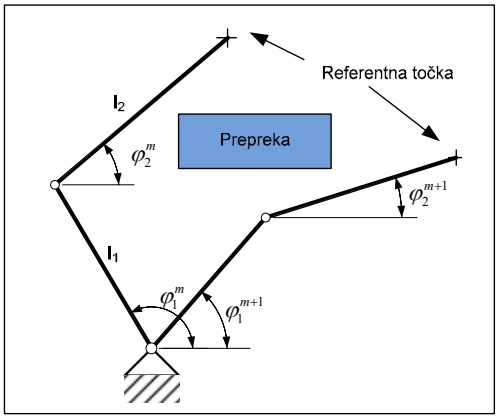
U primjeru s jednim robotom u radnom prostoru, razmatra se scenarij kada se u okolini nalazi stacionarna prepreka. Zadatak je pronaći niz diskretnih konfiguracija robota koje ne rezultiraju kolizijom s preprekom, istodobno zadovoljavajući ostale već navedene kriterije.

Konfiguracije robota u početnom, $t = 0$ trenutku vremena i konačnom, $t = konačno$ trenutku zadane su. Evolucijskim algoritmom traže se među-konfiguracije koje definiraju tranziciju stanja $t \to t+1$. Pitanje koje se nameće, a vezano je s vremenskom složenosti složenost EA, kako je pokazano u odjeljku 2.5, je koliko je među-konfiguracija potrebno za realistično opisivanje ovog problema. Svako među-stanje biti će jedan dio kromosoma, koji će opisivati gibanje robota i predstavljati rješenje u prostoru odluke. Potrebno je odabrati takav broj među-stanja, da performanse EA budu u granicama prihvatljivosti, a da se iz niza tranzicija može vjerno rekonstruirati gibanje robota. Kada bi kromosomi bili, u najjednostavnijem slučaju, duljine $l = 2$, značilo bi da je kodirano samo jedno među stanje i kromosom bi imao oblik $\left\{\varphi_1^{t=t+1}, \varphi_2^{t=t+1}\right\}$ u prostoru $\mathcal{G}$. Zadatak bi EA bio pronaći par kutova,

koji rješava zadani problem, što je u prostoru $\mathcal{F}$ prikazano slikom [Slika 3-2]. Iz oblika kromosoma vidljivo je da početna i konačna stanja nisu kodirana radi uštede računalnih resursa. Početno i konačno stanje se tijekom evolucije ne mijenjaju, tako da su sva razvijena stanja uvijek instance sheme oblika:

$$\left\{\varphi_1^{t=0}, \varphi_2^{t=0}, \#_1^{t=1}, \#_2^{t=1}, \ldots, \#_1^{t=konačno-1}, \#_2^{t=konačno-1}, \varphi_1^{t=konačno}, \varphi_2^{t=konačno}\right\}, \tag{16}$$

čime se opravdava izostavljanje navedenih stanja iz kromosoma.



Slika 3-2. Rješenje uz kodiranje kromosomom duljine *l* = 2.

Za primjer prikazan slikom [Slika 3-2], diskretizacija je gruba, te je o međustanjima poznato malo informacija. Ovaj bi slučaj mogao uzrokovati potrebu za iznosom brzine gibanja koji izlazi izvan granica omeđenosti stvarnog robota. Potrebno je u razmatranje uzeti iznos maksimalne brzine kojom se robot može kretati [67]. Pretpostavlja se da se robot kreće iz početne u konačnu u prvom slučaju tako da se giba samo donji segment robota, u tom slučaju je zakret prvog zgloba jednak $\Delta\varphi_1$. U drugom slučaju, kada se kreće samo gornji segment robota, zakret drugog zgloba iznosi $\Delta\varphi_2$. U oba je slučaja put što ga prelazi referentna točka jednak i iznosi ΔS. U tom slučaju, prema slici [Slika 3-3] vrijedi izraz:

$$\Delta\varphi_1 \cdot \left(l_1 + l_2\right) = \Delta\varphi_2 \cdot l_2 = \Delta S, \tag{17}$$

gdje su l_1 i l_2 duljine odgovarajućih segmenata robota. Ako se pretpostavi da je vrijeme uzorkovanja između dviju uzastopnih konfiguracija iznosa T, put što ga prelazi referentna točka je:

$$\Delta S = V_{\max} \cdot T, \tag{18}$$

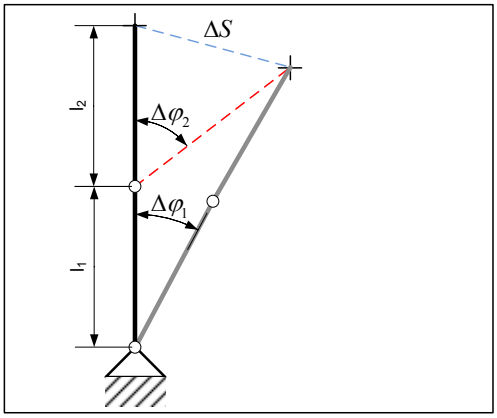
a pripadajući kutovi su:

$$\Delta\varphi_{1,\max} = \frac{V_{\max} \cdot T}{l_1 + l_2};\ \Delta\varphi_{2,\max} = \frac{V_{\max} \cdot T}{l_2}\ , \tag{19}$$

gdje je V_{max} najveća dopuštena brzina referentne točke, što je parametar koji je definiran od strane proizvođača robota. Uzme li se da je iznos V_{max} = 10 m/s te uz pretpostavku vremenskog intervala T = 0.01 s, uz $l_1 = l_2$ = 0.25 m, dobivaju se sljedeće vrijednosti kutova: $\Delta\varphi_1 \leq 11{,}5^\circ$ i $\Delta\varphi_2 \leq 23^\circ$, čime je omeđen diskretizacijski prostor i kromosomi bi trebali biti takvi da promjena kutova bude u izračunatim granicama. Do duljine kromosoma dolazi se razmatranjem kutova zakreta u zglobovima koje robot ostvaruje zakretom iz početne do konačne konfiguracije, uz pretpostavku da je funkcija kuta monotona, što je vrijednost definirana izrazom (26). Duljinu gena izračunava se pomoću sljedećeg izraza:

$$l \geq 2 \cdot \frac{\Delta\overline{\varphi}_i^G}{\Delta\varphi_{1,\max}},\quad l \in \mathbb{N} \wedge l \neq 0, \tag{20}$$

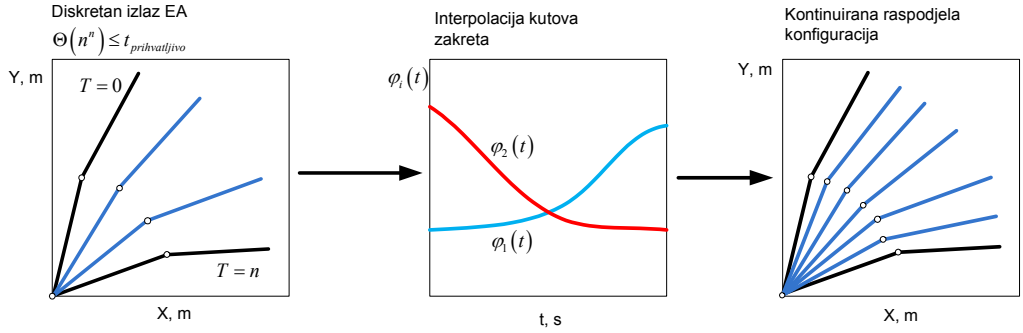
pri čemu je $\Delta\overline{\varphi}_i^G = \left|\varphi_i^{(n-2,G)} - \varphi_i^{(1,G)}\right|$ razlika između početnog i konačnog kuta zakreta *i*-tog segmenta robota, l je duljina kromosoma zaokružena na prvi veći prirodni broj, $\Delta\varphi_{1,\max}$ je maksimalni kut zakreta segmenta 1, izabran kao stroži kriterij od dva definirana izrazom (19). Brojka 2 uključena je jer kromosom kodira parove vrijednosti, za prvi i drugi segment. U najnepovoljnijem slučaju, kada bi kutovi bili odabrani na rubovima dopuštenog prostora, njihova bi razlika bila jednaka vrijednosti `p`, a pripadajuća duljina gena l = 32, što bi značilo da je prostor potrebno diskretizirati sa 16 među-konfiguracija u najnepovoljnijem slučaju.



Slika 3-3. Analiza kretanja referentne točke robota.

Ukoliko bi vrijeme *T* poprimilo manju vrijednost, pojavila bi se potreba za povećanom duljinom kromosoma. Kako je prikazano u poglavlju 2.5, gdje se govori o računskoj složenosti, za slučaj da algoritam ima karakteristiku eksponencijalnog karaktera, $\Theta\left(n^n\right)$, u tom se slučaju ne bi uopće moglo garantirati rješenje problema. Rješenje se u ovoj disertaciji predlaže tako da se duljina kromosoma drži manjom od neke kritične vrijednosti utvrđene eksperimentalno, za koju algoritam konvergira u polinomijskom vremenu, a do među-konfiguracija dolazi se interpolacijom kroz diskretan skup točaka, čiji je broj jednak duljini gena podijeljeno s dva, polinomom reda veće od dva. Na taj se način mogu rekonstruirati konfiguracije robota za proizvoljnu vrijednost *T*, uz konvergenciju algoritma u polinomijskom vremenu, kako je prikazano slikom [Slika 3-4].

U tom slučaju, kada postoji *prazan prostor* između dviju uzastopnih konfiguracija, može se postaviti pitanje može li robot uopće iskoristiti taj prazan prostor da se iz konfiguracije *T* rekonfigurira u konfiguraciju $T + 1$. Ovo se pitanje formalno razmatra kasnije u disertaciji za dva robota koji dijele radni prostor i dokazuje se da je uvijek moguće, uz određene geometrijske i kinematičke pretpostavke pronaći među-konfiguraciju koja rezultira gibanjem bez kolizije kroz *prazni* prostor.



Slika 3-4. Rekonstrukcija konfiguracija s po volji finom diskretizacijom *T*. Iz diskretnog izlaza EA, odabranog tako da se konvergencija ostvaruje u polinomijskom vremenu, interpolacijom odgovarajućim polinomom dolazi se do kontinuirane raspodjele kuta svakog zgloba ovisno o vremenu. Iz ove je raspodjele moguće rekonstruirati konfiguracije u proizvoljnom trenutku *T*.

Gornja slika predstavlja slijed rekonstrukcije kretanja robota iz grubo diskretiziranog (lijevo), interpolacijom kutova zakreta u vremenu (sredina) do fino diskretiziranog kretanja robota tijekom vremena (desno).

3.1.1 Minimizacija prijeđenog puta

Kako je već napomenuto, obzirom na kompliciranu prirodu višekriterijskih optimizacijskih problema i s ciljem boljeg razumijevanja među-djelovanja optimizacijskih kriterija, pristupa se sekvencijalnom rješavanju izloženog problema. Pri tome je prvi slučaj onaj kada se u radnom prostoru ne nalazi prepreka, te se putanja optimira samo prema kriteriju ukupnog puta kojega prelazi referentna točka robota gibajući se od zadane početne do zadane konačne konfiguracije u prostoru.

Kodiranje se izvodi vektorima realnih vrijednosti, pri čemu parovi vrijednosti označavaju iznose zakreta kutova u zglobovima, u radijanima. Funkcija dobrote u ovom slučaju ima oblik:

$$\Phi_1 = \Delta d^G = \sum_{T=1}^{T=n} \sqrt{\left(x_T^G - x_{T-1}^G\right)^2 + \left(y_T^G - y_{T-1}^G\right)^2} \rightarrow \min, \tag{21}$$

gdje je Δd^G put što ga referentna točka prelazi od početne do konačne konfiguracije u generaciji G, T je diskretno vrijeme kojim je obuhvaćena i početna i konačna konfiguracija robota, n je duljina gena uvećana za dva, zbog uključenih početne i konačne konfiguracije, x i y su koordinate referentne točke izvršnog člana.

Funkcijom dobrote zbrajaju se odsječci po dijelovima linearne putanje što ju opisuje referentna točka. Veću šansu za reprodukciju i izbor u roditeljsku populaciju imaju ona rješenja koja rezultiraju kraćim putanjama. Prema kriteriju definiranom jednadžbom (21) optimalno rješenje je odsječak pravca koji spaja početni i konačni položaj referentne točke. Kodirani kromosom općenito je u generaciji G sljedećeg oblika:

$$\left[\left\{\varphi_1^{(T,G)},\ldots,\varphi_i^{(T,G)}\right\},\left\{\varphi_1^{(2T,G)},\ldots,\varphi_i^{(2T,G)}\right\},\ldots,\left\{\varphi_1^{((n-2)T,G)},\ldots,\varphi_i^{((n-2)T,G)}\right\}\right], \tag{22}$$

pri čemu je i broj stupnjeva slobode gibanja, $i = 2$, T je vrijeme uzorkovanja između dvije uzastopne konfiguracije koje je radi jednostavnosti odabrano $T = 1$ s. Početna inicijalizacija vrijednosti zakreta kutova u zglobovima izvodi se stohastički:

$$\varphi_l^{j\cdot T,0} = \text{rand}\langle 0,\pi\rangle \tag{23}$$

pri čemu je *rand* funkcija generatora slučajnih vrijednosti, (j = 1,…,n-2; l = 1,…i). Iz razmatranja su isključene vrijednosti kutova veće ili jednake vrijednostima 0 i p jer bi te vrijednosti značile koliziju segmenta robota s podlogom.

Koriste se tri varijacijska operatora: selekcija, mutacija i križanje koji su implementirani kako slijedi. Selekcija je temeljena na ruletnom pravilu i proporcionalna je dobroti pojedinca. Za križanje implementiran je mehanizam *standardnog uniformnog uzorkovanja*. Nakon križanja, najbolji potomci se odabiru i tvore novu populaciju, uz direktan prijenos određenog postotka prethodne populacije, koji se kopiraju na mjesta najlošijih članova trenutne populacije. Obzirom da je kodiranje realnim brojevima, mutacija se izvodi na način da se određena pozicija (*lokus*) u genu mijenja, ukoliko do mutacije dolazi prema sljedećem izrazu:

$$\varphi_i^{(j\cdot T,G+1)} = \varphi_i^{(j\cdot T,G)} + N\left(0,\frac{1}{\sqrt{\pi}}\right) \tag{24}$$

u generaciji G, pri čemu je $N(\mu,\sigma)$ funkcija Gaussove razdiobe s očekivanjem `m` i standardnom devijacijom `s`.

Moguće je odmah uočiti da je promatrani problem spada u skupinu kontinuiranih problema s ograničenjima, odnosno da nisu sva rješenja, koja mogu nastati primjenom evolucijskih operatora prihvatljiva u prostoru `L`. Naime operator mutacije kako je definiran izrazom (24) može generirati rješenja koja na određenom lokusu imaju vrijednost kuta veću od najveće dopuštene. Obzirom da kodiranjem i mutacijom nije direktno moguće izbjeći ovaj problem, uvedena je metoda *popravljanja* koja se temelji na oduzimanju nasumične vrijednosti od vrijednosti trenutnog kuta tako da trenutni kut transformira u dopušteno područje.

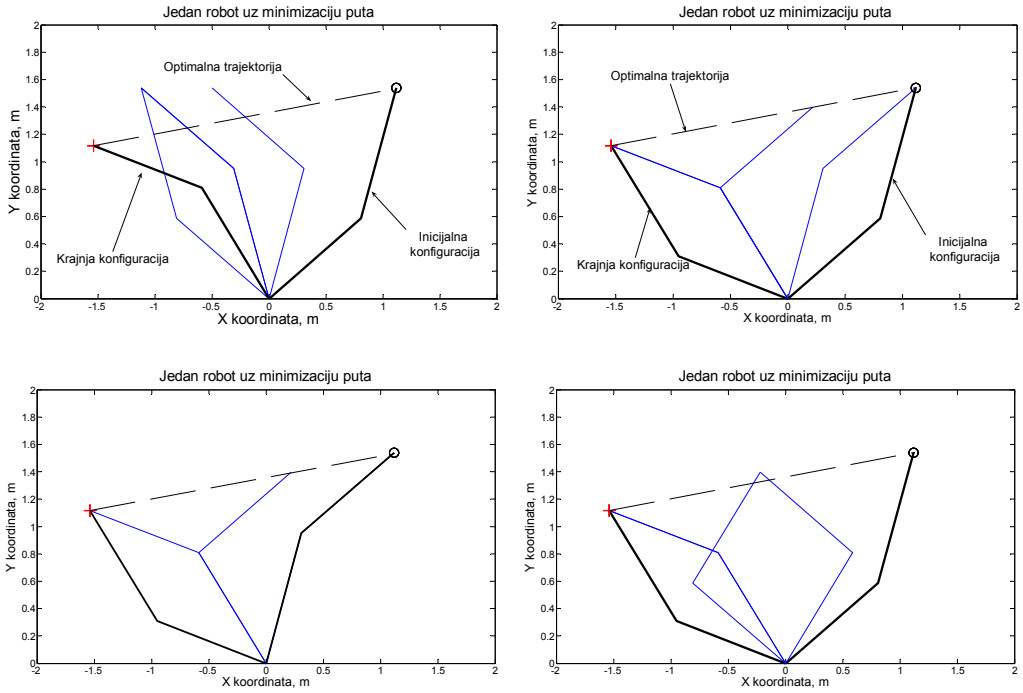
Tablica 3-1. Vrijednosti parametara EA za slučaj minimizacije puta.

Parametar	Vrijednost
Veličina populacije, N	20
Vjerojatnost križanja, p_m	0,7
Vjerojatnost mutacije	0,2
Metoda selekcije	*Ruletno pravilo*

Duljina kromosoma	$l = [6 – 12]$
Kriterij prestanka rada	*Hard limit*, G=200
Elitizam	*Da / Ne*

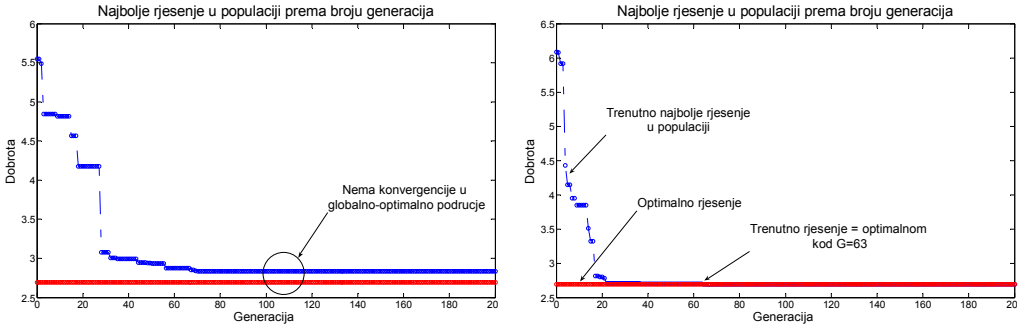
Tablica [Tablica 3-1] prikazuje skup parametara primijenjenog evolucijskog algoritma. Kod onih parametara gdje je kod vrijednosti naznačeno više elemenata, u daljnjem tekstu će se specificirati o kojem se točno parametru radi.

Na slici [Slika 3-5] prikazana su četiri rješenja pronađena evolucijskim algoritam u opisanom slučaju. Algoritam konvergira vrlo brzo, uz svega $G \approx 50$ generacija.



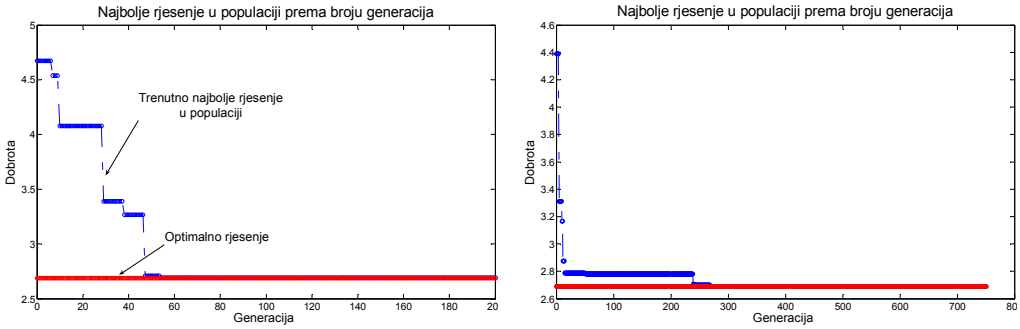
Slika 3-5. Rješenja pronađena EA u slučaju minimizacije prijeđenog puta.

Iako su tri od četiri prikazana rješenja dosegnula globalni optimum ili njemu vrlo blisko područje, rješenja nisu prihvatljiva. Naime, iako su kromosomi duljine između šest do dvanaest, algoritam daje rješenja koja u među-stanjima imaju dvije ili tri konfiguracije. To je posljedica činjenice da se neke među-konfiguracije međusobno preklapaju. To bi kod realne primjene imalo za posljedicu čekanje robota u određenoj konfiguraciji, a zatim naglo kretanje u sljedeću konfiguraciju koja se ne preklapa s prethodnom, što bi rezultiralo trzajima u radu.



Slika 3-6. Konvergencija EA u slučaju minimizacije prijeđenog puta.

Za konfiguracije prikazane slikom [Slika 3-5], slikama [Slika 3-6] i [Slika 3-7] prikazano je ponašanje algoritma prilikom konvergencije. Prema izgledu ovih dijagrama zaključuje se da algoritam nema poteškoća pri identifikaciji globalnog optimuma, gradijent je izražen. Međutim primjećuje se svojstvo algoritma da je u početnoj fazi u stanju napraviti značajan napredak, dok za fino ugađanje rješenja, kada je rješenje već blizu optimuma zahtijeva veći broj generacija. Ovo je posljedica stohastičke prirode operatora EA, i objašnjeno je u literaturi kao *exploration – exploitation* odnos [42].



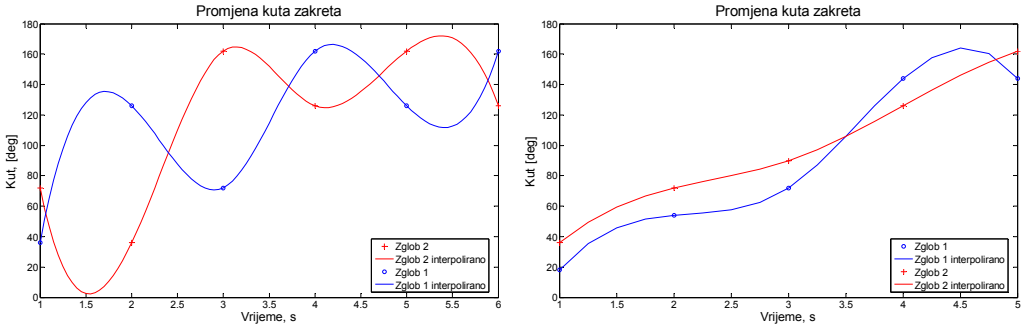
Slika 3-7. Nastavak prikaza konvergencije EA u slučaju minimizacije prijeđenog puta.

Obzirom na stohastičku prirodu EA, moguć je slučaj da se čak niti u ovako jednostavnom slučaju, u nekom unaprijed određenom broju generacija ne dosegne optimum, što je prikazano desno na slici [Slika 3-6]. Iako bi u ovom slučaju algoritam vjerojatno u konačnom broju iteracija dosegao globalni optimum, ovo je osobina iz koje proizlaze mnoge teškoće, kako pri karakterizaciji EA, tako i pri njihovoj implementaciji.

Vezano uz problem koji se u ovoj disertaciji rješava, moguć je slučaj da se nakon određenog broja generacija ne pronađe takav skup trajektorija koje rezultiraju gibanjima bez kolizije za oba robota. Obzirom da se zna da takva rješenja nisu prihvatljiva, jednostavno je

implementirati u algoritam metodu za odbacivanje takvih rezultata evolucije, primjerice ponovnim pokretanjem programa ili promjenom određenog skupa parametara algoritma.

Na sljedećoj slici prikazana je ovisnost kutova zakreta $\varphi_1(t)$ i $\varphi_2(t)$, za dva slučaja pronađenih konfiguracija koje su prema opisanom problemu konvergirala u optimum.



Slika 3-8. Ovisnost kuta zakreta o vremenu za dva slučaja konvergencije.

Na slici lijevo vidljivo je značajno kolebanje, koje je posljedica redundantnosti probleme, te iako je putanja optimalna po kriteriju prijeđenog puta, ovaj niz konfiguracija nije prihvatljiv za implementaciju na realnom sustavu. Na realnom robotu ovo bi imalo za posljedicu nekontrolirano i nepotrebno mijenjanje konfiguracija robota iz *lijeve* u *desnu* konfiguraciju i obratno. Povrh toga, prisutne su nagle promjene akceleracije što nikada nije poželjno kod programiranja i oblikovanja realnih robotskih sustava jer značajno nepovoljno utječe na vijek trajanja robota. Značajno bolje ponašanje primjećuje se na slici [Slika 3-8], lijevo. Diskretne vrijednosti kutova zakreta interpolirane su polinomom trećeg reda. Obzirom da ovako dobivena ovisnost predstavlja kontinuiranu raspodjelu kuta u vremenu, potrebno je osigurati derivabilnost prve i druge derivacije funkcije promjene ovisnost kuta zakreta u zglobovima o vremenu.

Iz provedene analize može se zaključiti da predloženi EA uspješno rješava problem, iako funkcija dobrote nije adekvatno oblikovana, odnosno ne uzima u obzir skup parametara dostatnih za evoluiranje zadovoljavajućih nizova konfiguracija. Funkciju dobrote potrebno je proširiti dodavanjem dodatnih kriterija. Time problemska domena postaje složenija, ali i ponašanje razvijenog agenta prilagođeno je toj, sada kompleksnijoj domeni.

3.1.2 Minimizacija kuta rotacije u zglobovima

Nepotrebno kolebanje redundantnog sustava između lijeve i desne konfiguracije robota nepovoljno je s više aspekata. Prvenstveno zbog bespotrebnog utroška energije i vremena, ali

i trošenja robota. Ova pojava može se ukloniti ukoliko se u funkciju dobrote uključi parametar koji evaluira ukupan kut rotacije u zglobovima, pri tranziciji robota od početne do konačne konfiguracije. Taj kut treba biti minimalan, te se minimizacijom odgovarajućeg dijela funkcije dobrote omogućuje evolucijskom algoritmu pronalaženje konfiguracija koje zadovoljavaju navedeni kriterij.

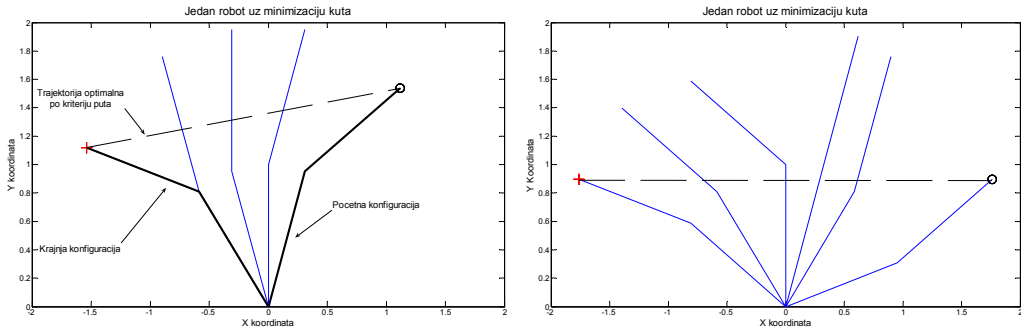
Kut što ga članak *i* prelazi u generaciji *G*, od poznate početne do poznate konačne konfiguracije određen je izrazom:

$$\Phi_2 = \Delta\varphi_i^G = \sum_{T=1}^{T=n} \left| \varphi_i^{(T,G)} - \varphi_i^{(T-1,G)} \right| \rightarrow \min. \tag{25}$$

U optimalnom slučaju, ukupna vrijednost promjene kuta rotacije članka *i* jednaka je:

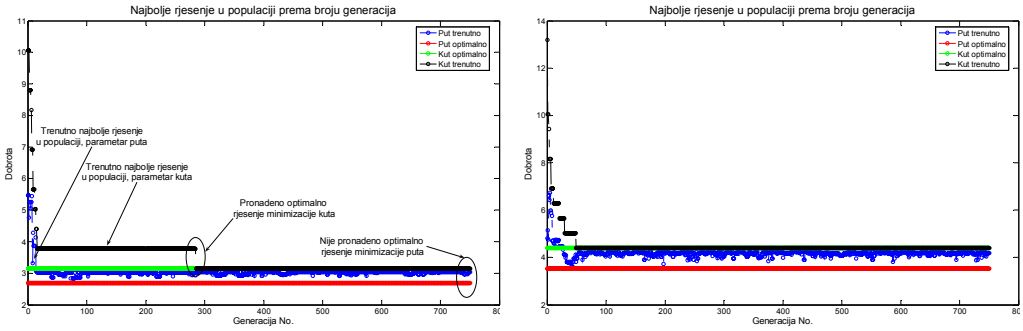
$$\Delta\overline{\varphi}_i^G = \left| \varphi_i^{(n,G)} - \varphi_i^{(1,G)} \right|, \tag{26}$$

što bi značilo da niti u jednom trenutku ne dolazi do promjene smjera zakreta u zglobovima. Optimum ovog kriterija je dohvatljiv ukoliko u okolini ne postoje prepreke, te se robot može nesmetano, bez odmicanja, uvijek pomicati između dvije uzastopne konfiguracije.



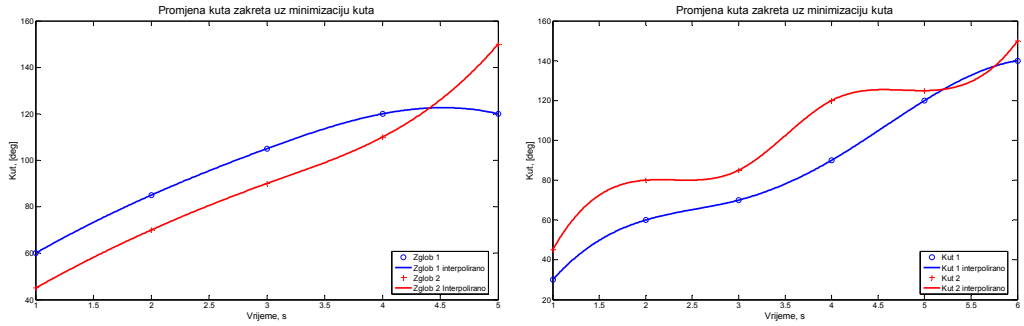
Slika 3-9. Rješenja pronađena EA u slučaju minimizacije kuta.

Slikom [Slika 3-9] prikazana su dva rješenja kada se kao kriterij koristi samo minimizacija kuta, kako je definirano izrazom (25). Algoritam konvergira u optimum vrlo brzo, prosječan broj generacija $G \approx 100$. Za dva slučaja prikazana slikom [Slika 3-9], konvergencija ovisna o broju generacija algoritma prikazana je [Slika 3-10].



Slika 3-10. Konvergencija EA u slučaju minimizacije kuta.

Na slici je, osim konvergencije kriterije kuta, prikazana i promjena kriterija puta o broju generacija.



Slika 3-11. Promjena kuta zakreta u zglobovima u vremenu.

Iako se kriterij puta što ga prelazi referentna točka ne evaluira funkcijom dobrote, primjećuje se da kriterij puta postiže stabilnu, ali lokalno optimalnu vrijednost. Iz ovog ponašanja može se zaključiti da postoji veza između kriterija puta i kriterija puta.

Slikom [Slika 3-11] prikazana je promjena kutova zakreta ovisna o vremenu za segmente robota. Na prikazu lijevo vidljiv je način promjene kutova bez promjene trenda, odnosno konstantan je prirast kuta u pozitivnom smislu. Kod slike desno, primjećuje se također stalan rast, uz malo manje jednoliko ponašanje što je posljedica nejednolikosti raspodjele kutova u vremenu, odnosno, prirast kuta nije konstantan. Odgovarajuće derivacije polinoma prikazanih gornjom slikom predstavljaju kutnu brzinu i kutno ubrzanje robota, o čemu je potrebno voditi računa pri izboru intepolacijskog polinoma.

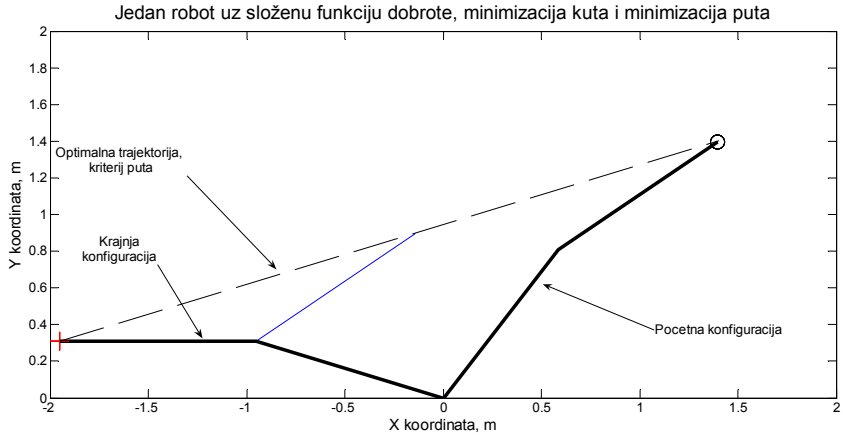
3.1.3 Višekriterijska funkcija dobrote, minimizacija puta i minimizacija kuta

Iz prethodnih je razmatranja vidljivo da pristupi kod kojih se pojedinci iz populacija vrednuju ili kriterijem ukupno prijeđene udaljenosti referentne točke, ili kriterijem zakreta kuta u zglobovima ne dovode do rješenja problema na zadovoljavajući način. Naime, ovako dobivene konfiguracije robota nisu primjenjive na stvarnom robotu. U slučaju minimizacije kuta, implementacija je čak i moguća, ali dobivene putanje ne optimiraju adekvatno kriterij minimizacije puta te se javlja povećan utrošak vremena.

Kako bi se eliminirali ovi nepovoljni utjecaji, formira se složena funkcija cilja, koja sadrži dvije komponente, opisane izrazima (21) i (25). Funkcija cilja sada ima oblik:

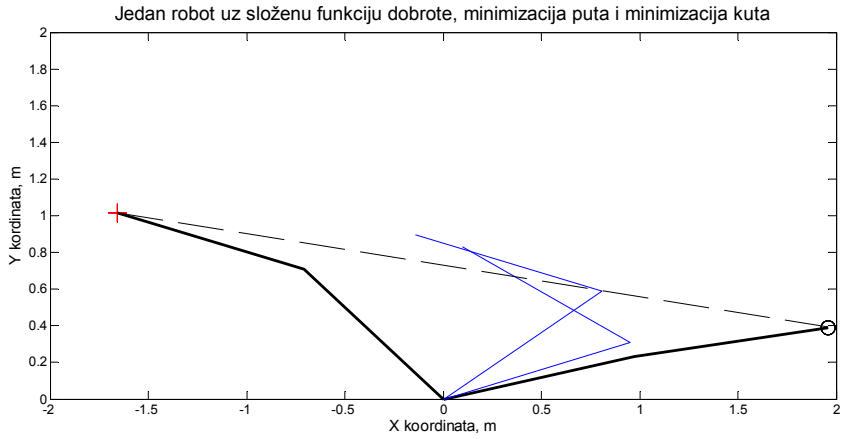
$$\Phi_{1,2} = w_1 \cdot \sum_{T=1}^{T=n} \sqrt{\left(x_T^G - x_{T-1}^G\right)^2 + \left(y_T^G - y_{T-1}^G\right)^2} + w_2 \cdot \sum_{T=1}^{T=n} \left|\varphi_i^{(T,G)} - \varphi_i^{(T-1,G)}\right| \rightarrow \min, \quad (27)$$

pri čemu su w_1 i w_2 težinski faktori kojima je moguće dati naglasak određenom kriteriju, te vrijedi $w_1 + w_2 = 1$, dok je smisao ostalih veličina kako je opisano izrazima (21) i (25). Ovako definiranom funkcijom cilja omogućuje se istodobna optimizacija po kriterijima prijeđenog puta i kuta rotacije u zglobovima.



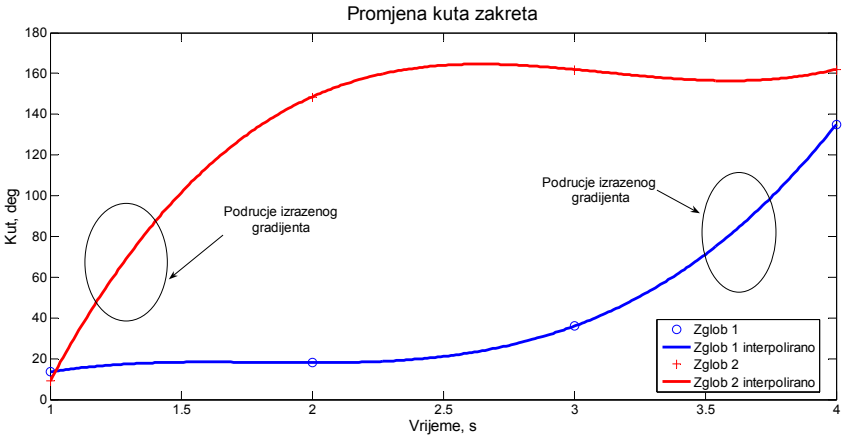
Slika 3-12. Rješenje EA uz složenu optimizaciju prijeđenog puta i ukupnog kuta rotacije, slučaj I.

S gornjih slika je vidljivo, da iako je duljina kromosoma l = 8 u oba slučaja, većina konfiguracija je grupirana ili jedne preko drugih ili ispod početne i konačne točke. iako je sada promjene kuta monotona funkcija što je preduvjet za njezinu konvergenciju, javlja se nepovoljan efekt velikih gradijenata, odnosno prirast kuta u jedinici vremena je nedopustivo velik, kako je vidljivo na slikama [Slika 3-14] i [Slika 3-15]. U oba slučaja prirasti kuta prelaze maksimalno dopuštenu vrijednost od 11,5°.



Slika 3-13 Rješenje EA uz složenu optimizaciju prijeđenog puta i ukupnog kuta rotacije, kriterij puta nije optimalno zadovoljen, slučaj II.

Konvergencija algoritma je vrlo dobra sa zadovoljavanjem optimalne vrijednosti ona kriterija za $G \approx 200$ generacija algoritma.

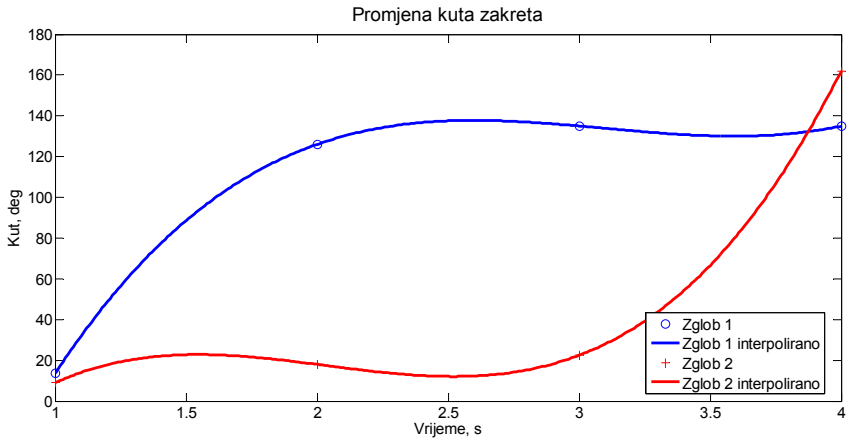


Slika 3-14. Ovisnost kutova zakreta o vremenu za slučaj I.

Konvergencija algoritma za dva prikazana slučaja ilustrirana je slikama: [Slika 3-16] i [Slika 3-17]. U oba slučaja algoritam nema poteškoća s konvergencijom, iako je za slučaj II kriterij optimizacije puta bio vrlo blizu optimalne vrijednosti koja nije dosegnuta. Opet se primjećuje svojstvo algoritma koji je u vrlo ranoj fazi, kod približno dvadesete generacije napravio značajan napredak, te je za daljnje fino ugađanje parametara potreban veći broj iteracija koje vode u konačno optimalno stanje.

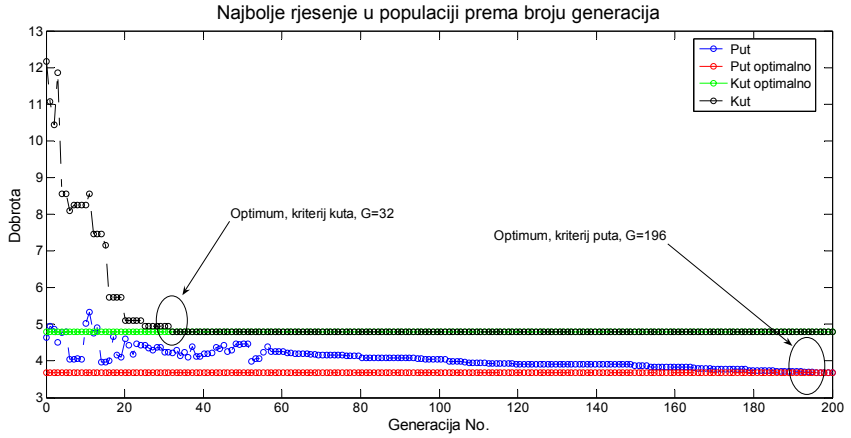
U prethodnoj diskusiji je zaključeno kako dva razmatrana kriterija ne daju zadovoljavajuće rezultate zbog izraženog prirasta kuta, pri čemu prirast izlazi izvan dopuštenih granica što bi

značilo da u stvarnosti robot sa omeđenim iznosima brzine i akceleracije ne bi mogao u raspoloživom vremenu dohvatiti referentne točke.



Slika 3-15. ovisnost kutova zakreta o vremenu za slučaj II.

Potrebno je uključiti dodatni kriterij, kojim će se uzeti u obzir ravnomjernost raspodjele prolaznih položaja referentne točke. Dodavanjem ovog kriterija očekuje se da će gradijent ostati u dopuštenom intervalu te da će problem planiranja putanje u slučaju kada u radnom prostoru nema prepreke skupom od ova tri kriterija biti riješen.



Slika 3-16. Konvergencija EA, slučaj I.

3.1.4 Višekriterijska funkcija dobrote; minimizacija puta, minimizacija kuta i minimizacija neravnomjernosti profila brzine

Poželjno je da referentna točka na izvršnom članu robota prelazi stalnu udaljenost u jedinici vremena. Posljedica toga je ravnomjeran profil brzine referentne točke, bez naglih promjena,

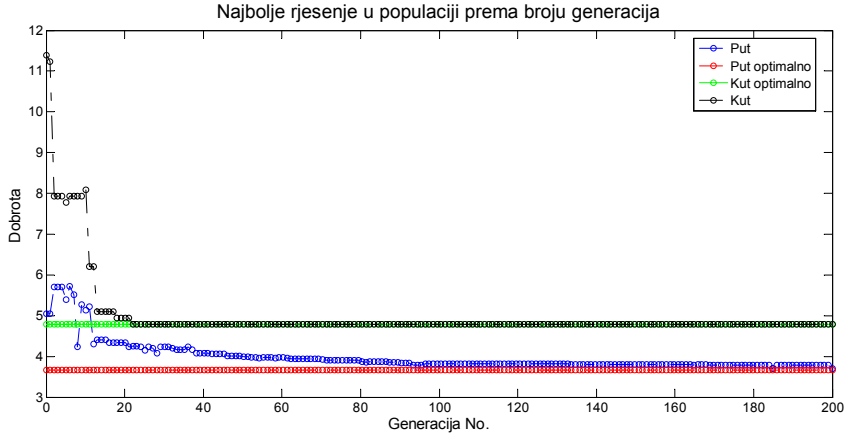
trzaja u zglobovima. Udaljenost koju vrh robota prijeđe u generaciji G, u dvije uzastopne konfiguracije, T i $T+1$ može se izračunati pomoću sljedećeg izraza:

$$\Delta d^G = \sum_{T=1}^{T=n} \sqrt{\left(x_T^G - x_{T-1}^G\right)^2 + \left(y_T^G - y_{T-1}^G\right)^2}, \tag{28}$$

funkcijom dobrote potrebno je vrednovati ravnomjernost gornjeg izraza, što se može na sljedeći način:

$$\Phi_3 = \Delta d_{\max}^G - \Delta d_{\min}^G \to \min, \tag{29}$$

pri čemu su vrijednosti $\Delta d_{\max}^G$ i $\Delta d_{\min}^G$ najveća odnosno najmanja udaljenost između dva uzastopna položaja referentne točke na izvršnom članu robota izračunata izrazom (28). U optimalnom slučaju, kada su sve udaljenosti koje referentna točka prelazi u jediničnom vremenu jednake, vrijednost izraza (29) poprima vrijednost 0, te je kriterij minimizacije neravnomjernosti profila brzine zadovoljen. Sam kriterij neravnomjernosti nije dostatan za planiranje putanje jer se u izrazu (29), kojim se evaluiraju pojedinci populacije ne nalaze informacije o početnom i konačnom položaju robota te se primjenom samo ovog kriterija ne dobivaju smisleni rezultati planiranja putanje.



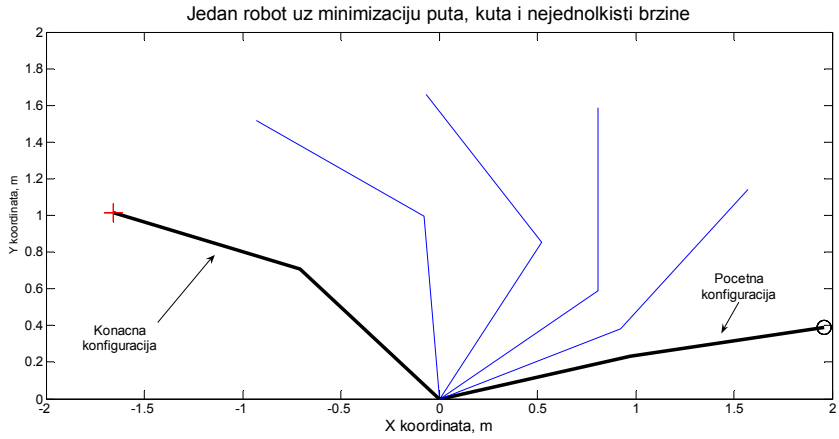
Slika 3-17. Konvergencija EA, slučaj II.

Stoga se odmah prelazi na definiranje složene funkcije dobrote koja je oblika:

$$\begin{aligned}\Phi_{1,2,3} = {} & w_1 \cdot \sum_{T=1}^{T=n} \sqrt{\left(x_T^G - x_{T-1}^G\right)^2 + \left(y_T^G - y_{T-1}^G\right)^2} + w_2 \cdot \sum_{T=1}^{T=n} \left|\varphi_i^{(T,G)} - \varphi_i^{(T-1,G)}\right| + \\ & + w_3 \cdot \left|\Delta d_{\max}^G - \Delta d_{\min}^G\right| \to \min,\end{aligned} \tag{30}$$

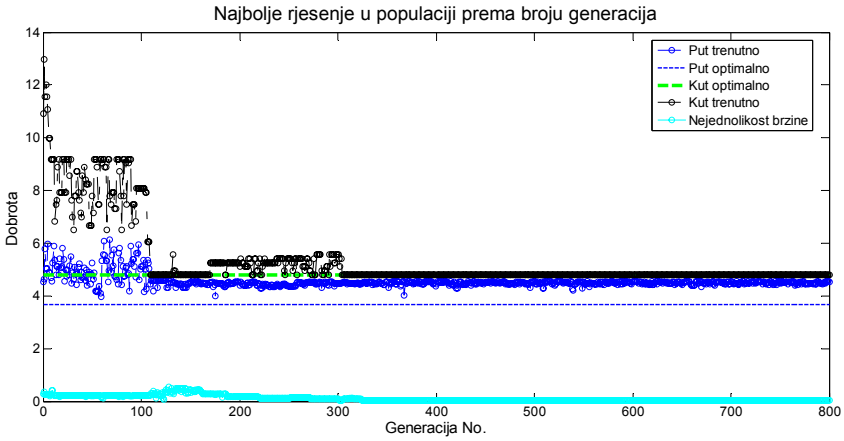
kojom se istodobno evaluiraju tri objašnjena kriterija. Smisao svih veličina istovjetan je onom iz izraza (21) - (29).

Slikom [Slika 3-18], prikazano je jedno rješenje dobiveno evolucijskim algoritmom. U odnosu na rezultate dobivane funkcijama dobrote koje nisu uključivali kriterij ravnomjernosti profila brzine, ovako dobivena rješenja pokazuju puno bolja svojstva. Obzirom da prostor odluke sada ima tri dimenzije, za pronalaženje prihvatljivog rješenja potrebno je uložiti znatno veće računalne resurse, $G \approx 800$. Pri tome je za pronalaženje globalnog optimuma, čak i u ovako jednostavnom slučaju općenito potrebno uložiti i trostruko veći broj generacija. Stoga se prihvaća kompromisno rješenje koje se postiže kada vrijednosti parcijalnih funkcija cilja Φ_1, Φ_2, Φ_3 dosegnu preddefiniranu prihvatljivu vrijednost.



Slika 3-18. Rješenje EA uz složenu funkciju dobrote, kriteriji prijeđenog puta, kuta zakreta i profila brzine.

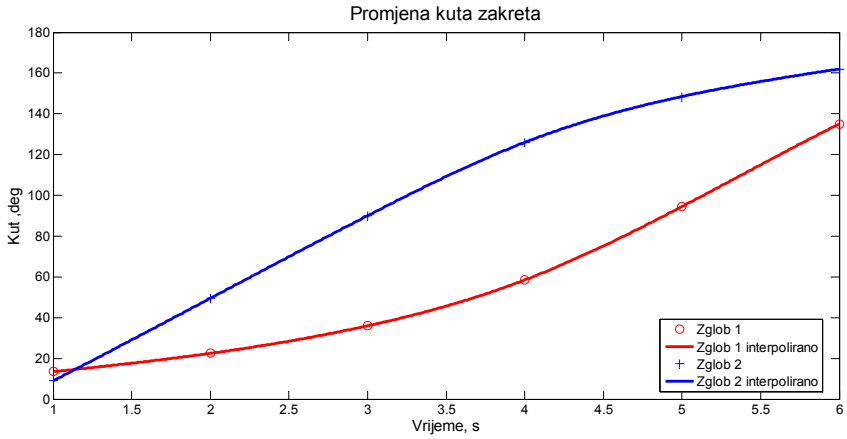
Na slici Slika 3-19 prikazana je konvergencija parcijalnih funkcija dobrote u ovisnosti o broju generacija. Primjetno je da su sve funkcije dosegnule stabilne vrijednosti, s time da su kriteriji minimizacije kuta i minimizacije nejednolikosti brzine dosegnuli i globalno optimalne vrijednosti, dok je kriterij minimizacije puta dosegnuo suboptimalnu, ali vrijednost vrlo blisku globalnom optimumu. Naime, da bi referentna točka pratila idealno pravac koji spaja početni i konačni položaj referentne točke, bile bi potrebne značajne rotacije u zglobovima, što je u suprotnosti s kriterijem minimizacije kuta rotacije.



Slika 3-19. Konvergencija EA za slučaj s tri minimizacijska kriterija.

Slika [Slika 3-20] vremensku ovisnost zakreta u zglobovima. Vidljivo je da je promjena ravnomjerno raspoređena na cijelom području, što je i bio cilj uvođenja dodatnog kriterija ravnomjernosti brzine.

Na taj način omogućuje se implementacija na realan robotski sustav, kako je prethodnim razmatranjem objašnjeno. U svim dijagramima koji prikazuju raspodjelu kuta zakreta u vremenu, početno vrijeme je definirano kao 1 radi naglašavanja da početna ($T = 0$), kao i konačna ($T = n$) konfiguracija nisu kodirane kromosomom i ne nalaze se na izlazu evolucijskog algoritma, iako njihovo uključivanje u grafički prikaz raspodjele kutova ne predstavlja problem. U tom bi se slučaju uključile dodatne dvije interpolacijske točke koje opisuju početnu i krajnju konfiguraciju robota. Što se tiče procesorskog vremena potrebnog za pronalaženje odgovarajućeg rješenja, simulacije su izvođene na prijenosnom računalu s 1 GB RAM-a, Intel Centrino Duo procesorom, s 2.16 GHz, te su vremena u slučaju optimizacije tri kriterija bile reda veličine 10 s.



Slika 3-20. Promjena kuta zakreta zglobova 1 i 2 uz minimizaciju prijeđenog puta, minimizaciju kuta zakreta i minimizaciju nejednolikosti brzine.

Iako predložena rješenja sada zadovoljavaju osnovni kriterij – kriterij implementacije na realnim robotskim sustavima, u razmatranje se uvodi slučaj kada u okolini postoji stacionarna prepreka konačnih dimenzija. Ovo razmatranje je dodatni korak prema konačnom razmatranju u ovoj disertaciji.

Dodatno, većina radova koja se bavi primjenom evolucijskih algoritama primijenjenih na sličnu domenu problema, razmatra ili problem kada u okolini prepreka ne postoji [68], [69], ili problem kada je prepreka točkasta, zanemarivih dimenzija [70], [71]. Na taj način problem se u velikoj mjeri pojednostavnjuje. U oba slučaja moguće je tražiti polinom koji zapravo predstavlja trajektoriju referentne točke izvršnog člana robota. U slučaju točkaste prepreke, potrebno je samo dodatno osigurati da se prepreka ne nalazi na dobivenom polinomu ili na samoj konfiguraciji robota u trenutku t_i. Ovakav je pristup neprimjenjiv u predloženoj disertaciji, gdje se razmatraju prepreke konačnih dimenzija, ili dva robota konačnih dimenzija. Sada je potrebno u svakom trenutku znati potpunu konfiguraciju robota, jer robot može bilo kojim segmentom uzrokovati koliziju, za razliku od pristupa koji se fokusira samo na referentnu točku.

3.1.5 Slučaj jednog agenta uz prisutnu statičku prepreku

Slučaj koji u obzir uzima prisutnost prepreke u radnome prostoru značajno komplicira prirodu problema. To se ogleda na sljedećim posljedicama: više nisu unaprijed poznati iznosi optimuma parcijalnih funkcija cilja; jasno suprotstavljeni kriteriji – izbjegavanje kolizija u suprotnosti je s kriterijem minimizacije prijeđenog puta primjerice; izračun kolizija uzima

značajno procesorsko vrijeme. Dakle, osim što dimenzionalnost problema i shodno tome, prostor rješenja raste, on postaje kompleksniji uslijed suprotstavljenosti kriterija i njegovo pretraživanje postaje teže. U slučaju kada je prepreka prisutna, očito je da ne postoji globalno optimalno rješenje koje će minimizirati sve parcijalne kriterije. Ako se prepreka nalazi u radnom prostoru robota, kao primjer može se uzeti slučaj kada prepreka leži na idealnoj trajektoriji – pravcu koji spaja referentnu točku na početnoj i konačnoj konfiguraciji, tada je upravo ta, optimalna trajektorija po kriteriju prijeđene udaljenosti referentne točke, neprihvatljiva jer bi rezultirala barem jednom kolizijom mjereno kriterijem evaluacije kolizija.

Smisao ovog kriterija je intuitivan, potrebno je na neki način kazniti one trajektorije koje rezultiraju kolizijama bilo kojeg segmenta robota s preprekom. Obzirom da se u ovom radu trajektorija uvijek diskretizira konačnim brojem međustanja i predstavlja nizom referentnih konfiguracija, nije dovoljna provjera kolizije samo u tim referentnim vremenskim intervalima. U tom bi se slučaju moglo dogoditi da iako niti jedna točka robota nikada ne dolazi u kontakt s preprekom, prepreka leži na području između dviju referentnih konfiguracija. Time bi se kolizija dogodila u interpoliranom trenutku, kada se događa promjena konfiguracije iz prethodnog u sljedeći trenutak t_i.

Prepreke su modelirane putem referentnog skupa točaka koje, spojene pravcima tvore područje koje je nedozvoljeno za pristup robota. Na ovaj se način mogu kreirati poligoni proizvoljne orijentacije i oblika.

Slučaj jedne statičke prepreke zapravo je poseban slučaj skupa problema kada se u prostoru nalazi dinamička prepreka, odnosno drugi robot. Za slučaj statičke prepreke, brzina prepreke jednaka je nuli tijekom cijele faze pomicanja robota. U slučaju dinamičke prepreke, moguće su dvije situacije: položaj prepreke mijenja se stohastički, ili slučaj kada se prepreka miče prema nekoj zakonitosti koju je moguće opisati. U slučaju kada je prepreka potpuno stohastička, planiranje trajektorije unaprijed teško je izvedivo. Potrebno je primijeniti čisto reaktivnu metodu upravljanja, opažajući trenutni položaj prepreke i odgovarajući gibanjem robota na to opaženo stanje. U slučaju kada postoje informacije o tome kakva bi zakonitost gibanja prepreke mogla biti, ove je informacije moguće iskoristiti za planiranje putanje jednog, ili u slučaju razmatranom u ovoj disertaciji, dva robota.

Kako je već navedeno, parcijalna funkcija cilja koja je zadužena za izbjegavanje kolizija treba promovirati raspodjelu po populaciji onakvih trajektorija koje imaju manji broj kolizija, a u optimalnom slučaju, broj je kolizija za trajektoriju jednak nuli.

Četvrta parcijalna funkcija dobrote, zadužena za vrednovanje putanja prema kriteriju broja kolizija definirana je sljedećim izrazom:

$$\Phi_4 = \sum_{T=1}^{T=n} C_T \rightarrow \min, \tag{31}$$

pri čemu je:

$$C_T = \begin{cases} 1, \text{ robot kolidira s preprekom} \\ 0 \text{ inače.} \end{cases} \tag{32}$$

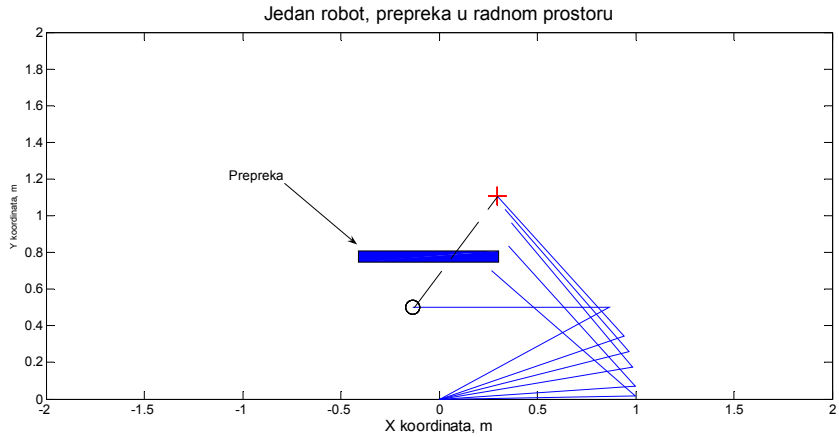
Kolizije se provjeravaju izračunom položaja sjecišta pravaca koje tvore bridovi prepreke i referentnog niza pravaca koji definira dvije uzastopne konfiguracije, što se formalno opisuje u poglavlju 3.2.7 gdje se formalno analizira model dvoručnog robota.

Normirana vrijednost funkcije dobrote $\tilde{\Phi}_4$ izračunava se tako da se vrijednost funkcije dobrote Φ_4 podijeli s ukupnim brojem kolizija cijele populacije. Svaki član populacije koji rezultira ne-nultom vrijednošću funkcije dobrote Φ_4 nije prihvatljivo rješenje na izlazu EA, prema definiciji problema planiranja kretanja postavljenoj u poglavlju 1.1. Potrebno je uvesti i težinski faktor w_4 pomoću kojega je moguće modulirati upliv svake od parcijalnih funkcija dobrote na konačno razvijeno rješenje. Ukupna funkcija dobrote sada ima oblik:

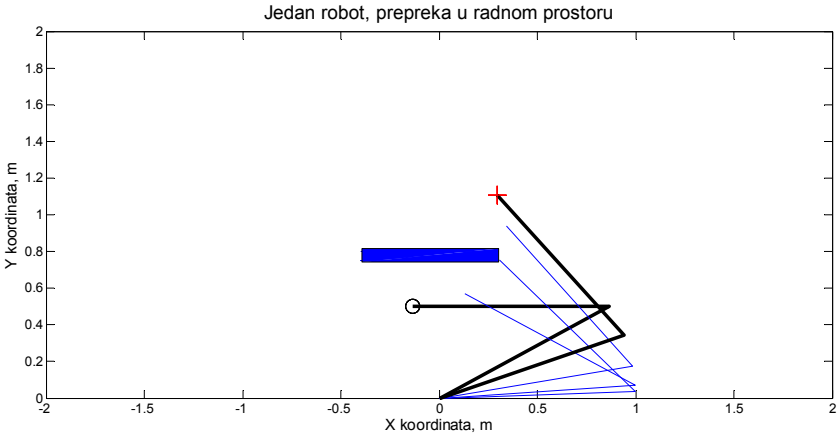
$$\begin{aligned} \Phi_{1,2,3,4} = w_1 \cdot \sum_{T=1}^{T=n} \sqrt{\left(x_T^G - x_{T-1}^G\right)^2 + \left(y_T^G - y_{T-1}^G\right)^2} + w_2 \cdot \sum_{T=1}^{T=n} \left|\varphi_i^{(T,G)} - \varphi_i^{(T-1,G)}\right| + \\ + w_3 \cdot \left|\Delta d_{\max}^G - \Delta d_{\min}^G\right| + w_4 \cdot \sum_{T=1}^{T=n} C_T \rightarrow \min, \end{aligned} \tag{33}$$

pri čemu je smisao svih veličina kako je prethodnim izrazima definirano. Funkcija dobrote iz izraza (33) zapravo predstavlja okolinu kojoj se agenti, koji su tom funkcijom evaluirani i vrednovani, moraju prilagoditi.

Jedan karakterističan slučaj, kada je odmicanje potrebno ilustrirati će se nizom slika koje slijede.

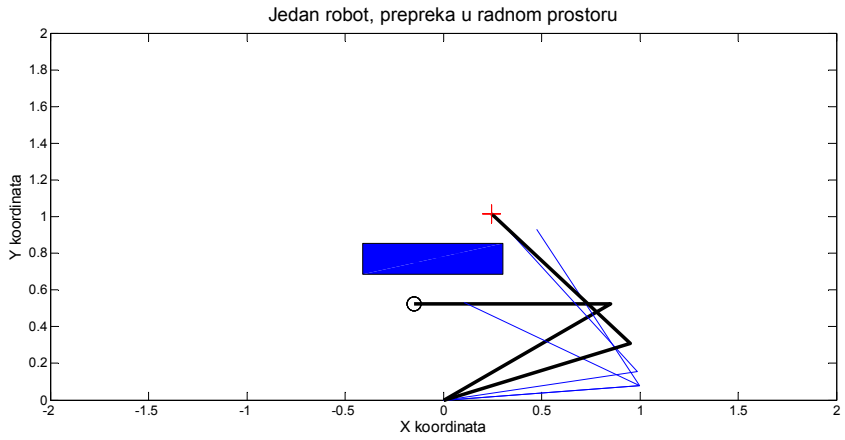


Slika 3-21. Rješavanje problema planiranja putanje uz jednu prisutnu prepreku, duljina gena *l*=8.

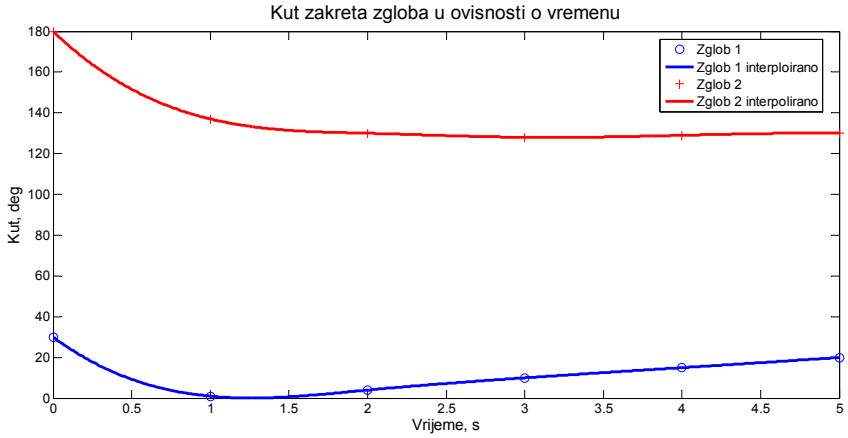


Slika 3-22. Rješavanje problema planiranja putanje uz jednu prisutnu prepreku, duljina gena *l*=6.

Ovako definirana funkcija dobrote propagira one agente, ona potencijalna rješenja koja rezultiraju manjim brojem kolizija, manjim prijeđenim putem referentne točke na izvršnom članu robota, manjim kutom zakreta u zglobovima i manjom neravnomjernošću profila brzine. Pitanje izbora težinskih faktora ovako definirane funkcije nije jednostavno i ne može se unaprijed odgovoriti. Potrebno je nizom kontroliranih simulacija eksperimentalno odrediti onakvu kombinaciju težinskih faktora koji daju najbolje rješenje problema, odnosno to je kombinacija s kojom u najvećem broju slučajeva algoritam daje zadovoljavajuće rješenje.



Slika 3-23. Suboptimalno rješenje na izlazu evolucijskog algoritma.



Slika 3-24. Kontinuirana promjena kuta zakreta zgloba *i* u vremenu za niz konfiguracijama prikazan slikom [Slika 3-21].

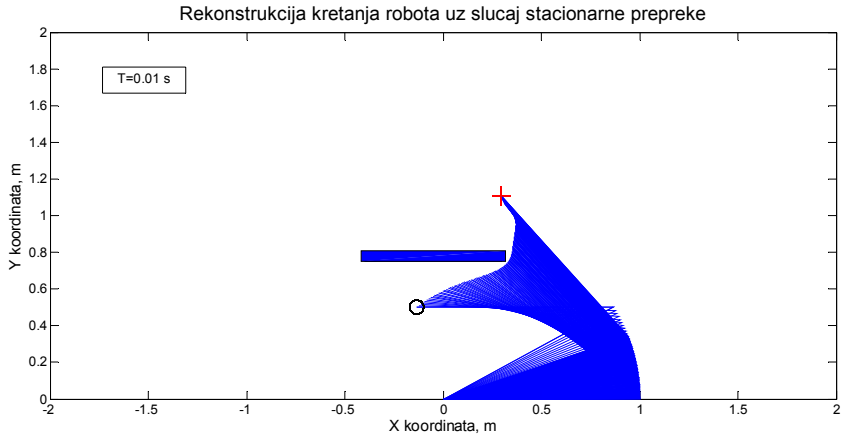
Simulacijama se pokazalo da težinski faktori nisu neovisni o okolini, te da je u određenim uvjetima, veći značaj potrebno dati faktoru w_4, što je slučaj kada je potrebno znatno odmicanje od trajektorije optimalne po kriteriju minimizacije puta. Upravo taj slučaj, kada je odmicanje koje je potrebno značajno je i najzanimljiviji slučaj pri proučavanju planiranja gibanja, ujedno i najteže rješiv, što će se kasnije u disertaciji i formalno pokazati. Naime, određeni se slučajevi, a to je prvenstveno slučaj kada prepreka na zauzima dio prostora koji je omeđen početnom i konačnom konfiguracijom prostora, mogu svesti na slučaj planiranja bez prepreke.

Gornjim slikama prikazana su dva rješenja za koja je problem planiranja uspješno riješen. Slika 3-21 prikazuje i optimalnu trajektoriju prema kriteriju prijeđene udaljenosti

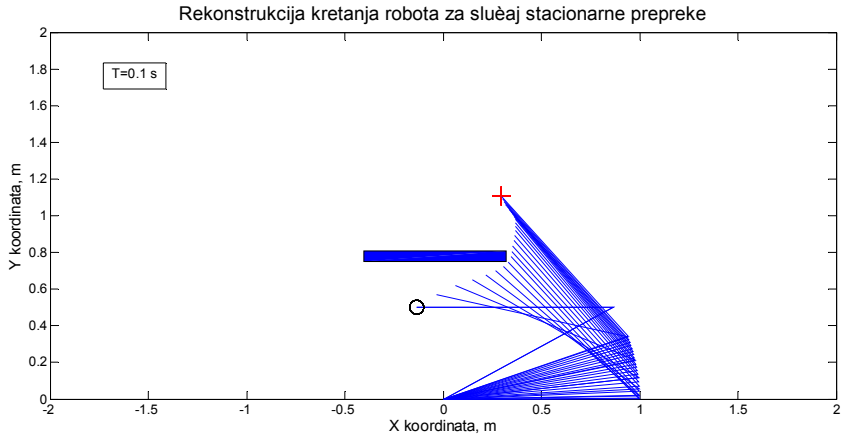
čime se ilustrira veza između dva kriterija, izbjegavanja kolizija i prijeđenog puta, koji su u suprotnosti jedan s drugim.

Obzirom da je slučaj s preprekom koja se nalazi u okolini potencijalno opasan, bilo da se radi o mogućem oštećenju robota pri koliziji ili o ozljedi čovjeka u najgorem slučaju, provest će se dodatna analiza ovog slučaja. Kao i u prethodnim slučajevima, moguće je doći do kontinuirane ovisnosti kuta zakreta o vremenu, kako je objašnjeno ranije. Ova se zavisnost prikazuje slikom [Slika 3-24] za rješenje EA koje je prikazano na slici [Slika 3-21.]

Iz gornje je slike moguće rekonstruirati putanje s po volji malim korakom diskretizacije. U prvom slučaju, slika [Slika 3-25] diskretizacija je izvršena s $T = 0.01$ s, dok je na slici [Slika 3-26] diskretizacija $T = 0.1$ s. Vidljivo je iz prikazanih slika da niti u međustanjima, koja nisu izravan izlaz evolucijskog algoritma nema kolizija s preprekom u radnom prostoru. Na ovaj način je moguće zadovoljiti uvjete na iznos maksimalnog porasta funkcije kuta zakreta u zglobu o vremenu.

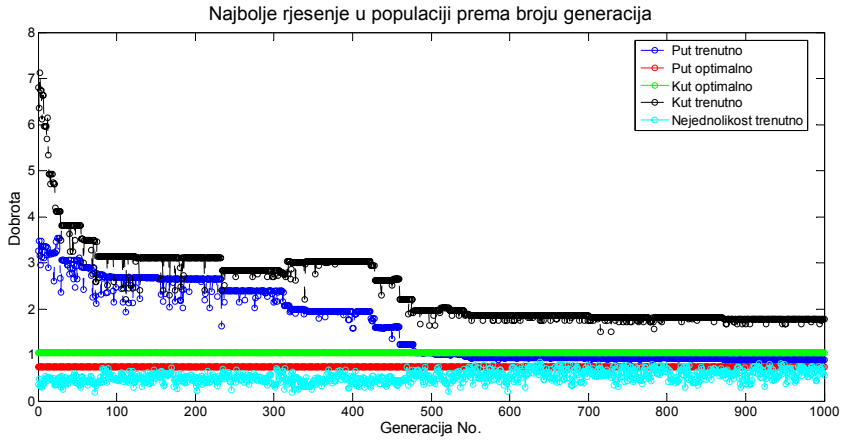


Slika 3-25. Kretanje robota rekonstruirano s vremenskim korakom $T = 0.01$ s.



Slika 3-26. Kretanje robota rekonstruirano s vremenskim korakom $T = 0.1$ s.

Konvergencija algoritma za isti slučaj prikazana je slikom [Slika 3-27]. U ovom slučaju nije moguće istodobno zadovoljiti sve kriterije na optimalan način. Na slici nije prikazana optimalna vrijednost kriterija nejednolikosti jer je on neovisan o konfiguracijskom prostoru i uvijek iznosi 0. U ovom je primjeru konačno rješenje pronađeno kod generacije $G \approx 500$, što je uz sklopovlje opisano u odjeljku 3.1.4 iznosilo približno 20 sekundi CPU vremena. Pri tome je vremenski najzahtjevnija operacija provjeravanja kolizije.



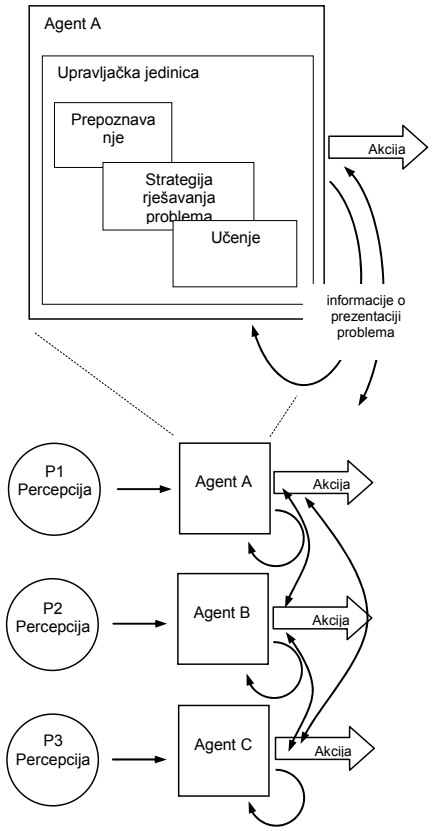
Slika 3-27. Konvergencija EA za rješenje prikazano slikom [Slika 3-22].

Prikazanim je razmatranjem pokazano kako je evolucijski algoritam implementiran na ponašanje jednog agenta s prisutnom statičkom preprekom. Definirani su kriteriji pomoću kojih je moguće razviti ovo vrlo složeno ponašanje. Diskutirana su međudjelovanja ovih kriterija. Kao sljedeći logičan korak i proširenje, nakon potvrde algoritma na ovom problemu,

javlja se potreba razvoja takvog algoritma kojim je moguće planirati putanje za dva agenta koji dijele radni prostor.

3.2 Dva agenta u radnom prostoru

U slučaju kada se dva agenta nalaze u radnom prostoru, i ako nema ograničenja postavljenih na dostupni prostor njihovog kretanja, oni predstavljaju međusobno prepreke koje se kreću.



Slika 3-28. Struktura općenitog višeagentskog sustava.

Obzirom da se ovdje razmatra slučaj pronalaženja trajektorija za svakog agenta, tada je moguće međusobno prilagođavati prostore stanja i na taj način pronaći optimirane trajektorije za svakog robota. Agenti se nalaze u okolini, koja se mijenja, te je potrebno pronaći takvu zakonitost gibanja koja rješava problem planiranja putanje u promjenjivim uvjetima.

3.2.1 Agent, višeagentski sustav

Svaki sustav koji se sastoji od više objekata koje se po određenim kriterijima može dekomponirati na osnovne dijelove može se smatrati višeagentskim sustavom [72]. *Agent* je

(računalni) sustav s dvije značajne karakteristike, prvo, barem je djelomično sposoban za *autonomno djelovanje* i drugo, sposoban je za *interakciju* s ostalim agentima.

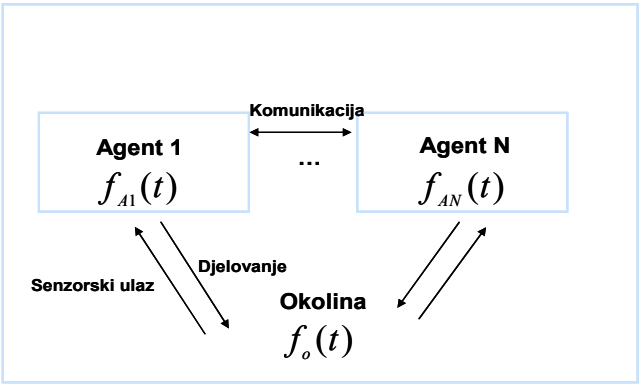
Definicija 10. ***Agent:*** je svaki sustav sposoban spoznati okolinu i djelovati na objekte u svojoj okolini.

Općenito se smatra da je agent minimalne konfiguracije sastavljen od najmanje jednog senzora i najmanje jednog aktuatora.

Definicija 11. ***Višeagentski sustav:*** je sustav sastavljen od najmanje dva *agenta* koji se nalaze u međusobnoj interakciji.

Današnja istraživanja u području umjetne inteligencije usmjerena su na koncept *racionalnog agenta*. Agent koji stalno pokušava optimirati svoje ponašanje koje se vrednuje odgovarajućom mjerom njegove uspješnosti, naziva se racionalnim agentom. Ova definicija je općenita i može uključivati ljudske agente (koji kao senzore imaju oči, a ruke zamjenjuju aktuatore), robotske agente (koji koriste kamere za percepciju i kotače kao aktuatore) ili računalne agente (koji imaju grafičko sučelje kao senzor i kao aktuator).

Općeniti višeagentski sustav prikazan je slikom [Slika 3-28]. Ilustrirane su veze koje su neophodne za komunikaciju među agentima te niz modula *učenje*, *rješavanje problema* i *prepoznavanje*, što su sve osobine agenta visoke razine. Primjenom višeagentskih paradigmi pojednostavnjuje se analiza i oblikovanje često veoma složenih struktura sastavljenih od više međudjelujućih sustava.



Slika 3-29. Agenti s pripadajućom funkcijom dobrote u uvjetima promjenjive okoline.

Slikom [Slika 3-29] prikazana je okolina čija je promjenjivost označena s $f_{\Omega}(t)$, a učinak djelovanja svakog agenta u toj okolini može se ocijeniti pripadajućom funkcijom

dobrote $f_{A_i}(t)$. Agenti su, po definiciji, sposobni primati informacije iz okoline te, u ovisnosti o primljenoj senzorskoj slici sa svojih ulaza i strukturu upravljačke jedinice djelovati na okolinu. Dodatno su, prema gornjem prikazu, agenti razmatrani u ovoj disertaciji sposobni za *komunikaciju*. Ovo je bitna karakteristika u promatranom primjeru planiranja putanje agenata, koja im omogućuje izmjenu informacija o trenutnoj konfiguraciju svakog agenta. Moguća su dva scenarija:

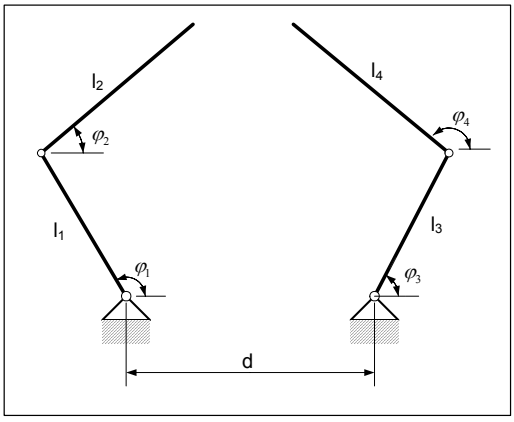
- direktna komunikacija, putem mrežnog TCP IP protokola, ili bežična komunikacija i
- posredna komunikacija, prepoznavanjem stanja ostalih agenata.

Pretpostavlja se mrežna komunikacija putem TCP IP protokola u daljnjem razmatranju.

Distribuiranim koevolucijskim algoritmom želi se pronaći takva kompozicija parcijalnih funkcija dobrote svakog agenta, kojom se optimira istodobno rad višeagentskog sustava kao cjeline prema skupu definiranih kriterija, c_i:

$$\exists F\left(f_{A_1},...,f_{A_N}\right)\forall t_i \rightarrow F_{opt} \mid c_1,...,c_n. \tag{34}$$

Pronalaženjem funkcije F uz odgovarajuće kriterije, rješava se problem planiranja putanje za svakog agenta posebno, ali i za višeagentski sustav kao cjelinu.



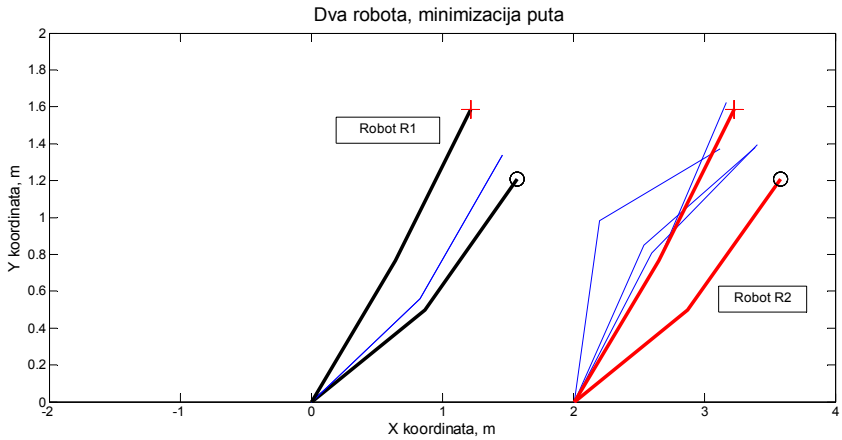
Slika 3-30. Dva agenta u radnom prostoru.

Slikom [Slika 3-30] prikazuje se višeagentski sustav dvoručne robotske konfiguracije čije se planiranje putanja rješava. Agenti su definirani kao dva 2R robota. Svaki agent ima dva segmenta, od kojih je jedan spojen za nepomičnu podlogu rotacijskim zglobom, dok je drugi

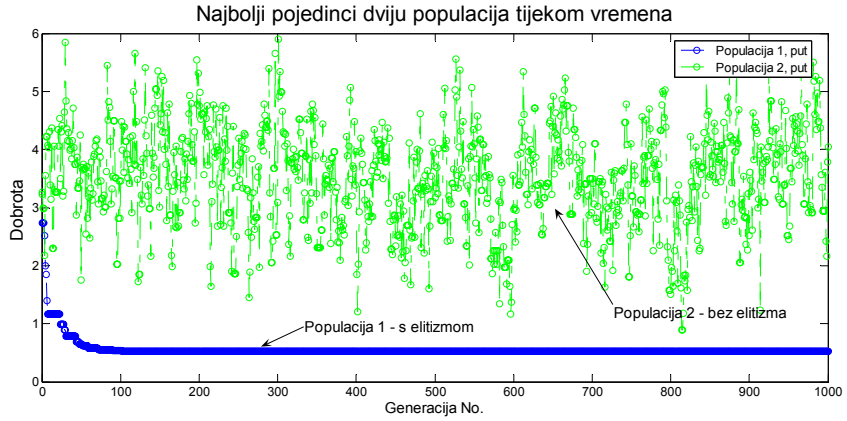
segment spojen za prvi sljedećim rotacijskim zglobom. Baze agenata međusobno su udaljene za iznos d. Duljina svakog segmenta označena je s l_i, dok je kut rotacije u odgovarajućem zglobu označen s φ_i.

3.2.2 Minimizacije duljine trajektorije

Kriteriji se i u slučaju dva agenta uvode sekvencijalno, analogno pristupu koji je proveden u razmatranju jednog agenta u radnom prostoru. Sve definicije evolucijskih algoritama, način kodiranja, selekcije, mutacije, inicijalizacije, iste su prethodno definiranim, osim kada je drugačije napomenuto.



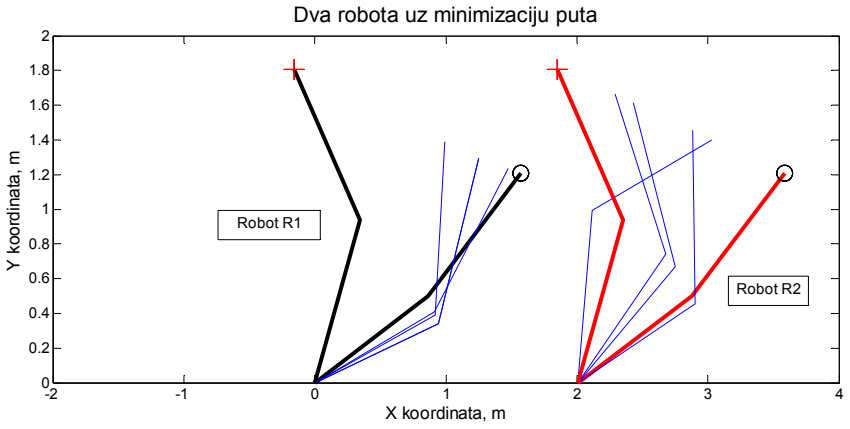
Slika 3-31. Evolucija dva robota u radnom prostoru. Lijevi robot, R_1 ima ugrađen elitizam i dostiže globalno optimalnu vrijednost.



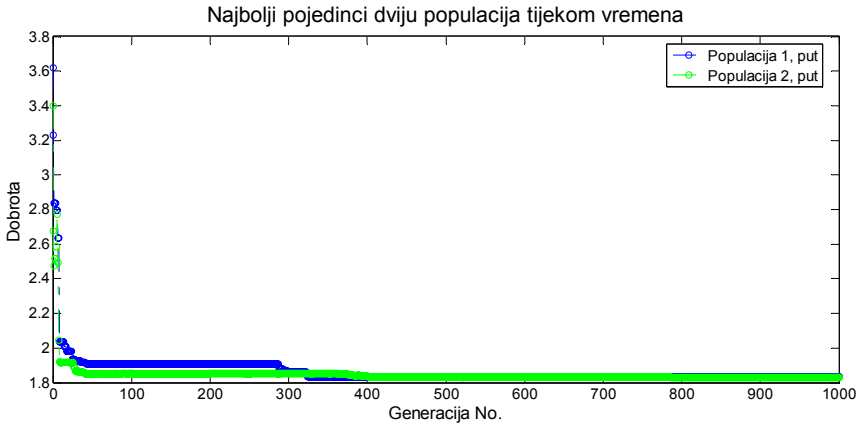
Slika 3-32. Konvergencija dviju populacija uz prisutnost / izostanak elitizma.

U slučaju kada se za dva robota u radnom prostoru u obzir uzme samo kriterij minimizacije puta potrebno je proširiti model jednog robota s poznatim početnim i krajnjim položajem drugoga robota. Također, u zadanim, početnim i krajnjim položajima ne smije biti kolizije, u suprotnom rješenje planiranja putanje ne postoji. U ovom slučaju, koevolucijski se algoritam svodi na osnovnu formu – paralelnog evolucijskog algoritma. Obzirom da se ponašanje svakog robota – agenta vrednuje duljinom trajektorije koju opisuje referentna točka na izvršnom članu, što je objektivna i unutrašnja mjera, prema definicijama 1. i 3., nema mogućnosti za primjenu koevolucijskog algoritma, prema definiciji 5.

Zadatak je riješen na način da su u početnoj fazi inicijalizirane dvije populacije po slučaju, koje se kasnije razvijaju potpuno neovisno jedna o drugoj. Iako bi bilo moguće implementirati populacije koje se razlikuju po svim bitnim kriterijima – veličini, vjerojatnosti križanja i mutacije, selekcijskoj metodi, uvjetu prekida itd. Obzirom da se radi o jednostavnom zadatku koji se uspješno rješava osnovnim algoritmom, ovi su parametri za obje populacije identični, osim parametra elitizma. U populaciju 1, koja je zadužena za evoluciju lijevog robota, implementiran je elitizam na način da se najboljim pojedincem iz prethodne iteracije zamjenjuje najlošiji pojedinac trenutne generacije. Pokazalo se da već ovaj jednostavni oblik elitizma ima povoljan utjecaj na konvergenciju, no, algoritam je u stanju pronaći optimalno rješenje ovog problema i bez elitizma uz dovoljno vrijeme na raspolaganju. Kod populacije 2, na kojoj nije implementiran elitizam vidi se znatno raspršenija kvaliteta trenutno najboljeg rješenja u populaciji. Ovo u nekim slučajevima može biti pogodno za izbjegavanje lokalno – optimalnog područja. Vidljivo je sa slike [Slika 3-31] da u slučaju desnog robota nije pronađeno optimalno rješenje, dok u slučaju lijevog robota je.



Slika 3-33. Evolucija dva robota u radnom prostoru, oba robota imaju ugrađen elitizam.



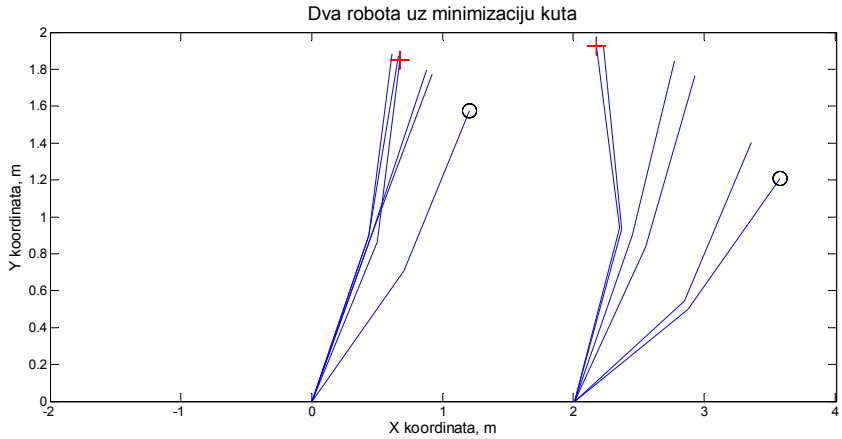
Slika 3-34. Konvergencija dviju populacija uz prisutnost elitizma kod obje populacije.

Slikama [Slika 3-33] i [Slika 3-34] prikazan je slučaj kada oba robota imaju ugrađen elitizam. Početna i konačna konfiguracija robota su iste za robot R_1 i za robot R_2. Ponašanje algoritama je gotovo simetrično, iako je rješenje koje daju različito na fenotipskoj razini, ono je s obziron na kriterij optimizacije istovjetno. U slučaju kada ova dva robota ne bi bila u interakciji, kada bi njihov razmak d bio dovoljno velik, ovo bi predstavljalo dva paralelna, neovisna algoritma implementirana na odvojenim upravljačkim računalima. U slučaju kada je d manji od kritične vrijednosti, ovaj se pristup može primijeniti samo uz uvjet da nigdje nema presjecišta površina koje zatvaraju početna i konačna konfiguracija robota R_1 i robota R_2.

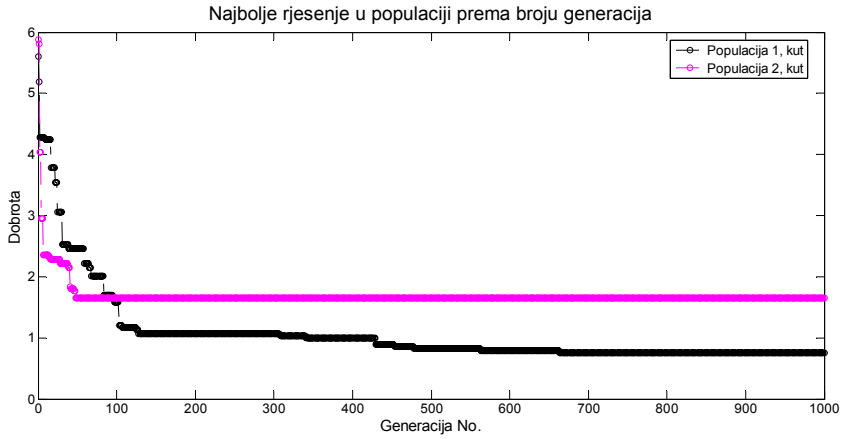
Analiza implementacije ostaje ista onoj provedenoj za jednog robota u radnom prostoru i neće se ovdje posebno provoditi.

3.2.3 Minimizacija kuta rotacije u zglobovima

Slučaj minimizacije kuta rotacije u zglobovima prikazan je slikama [Slika 3-35] i [Slika 3-36].



Slika 3-35. Evolucija dva robota u radnom prostoru, kriterij kuta rotacije u zglobovima.



Slika 3-36. Konvergencija populacija prema kriteriju puta.

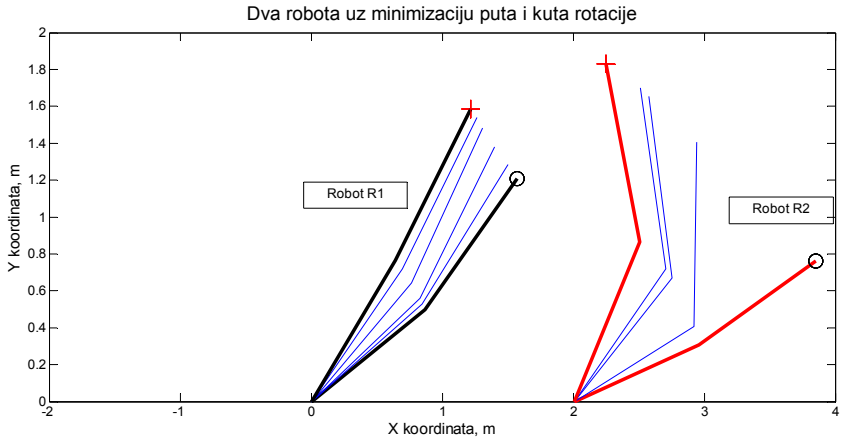
Učinak algoritma ne mijenja se u odnosu na situaciju kada je u radnom prostoru prisutan samo jedan robot evaluiran istim kriterijem. Iako u prikazanom slučaju populacija 1 koja predstavlja lijevog robota, R_1 nije pronašla globalni optimum, algoritam bez većih poteškoća pronalazi optimalno rješenje. Obzirom da su evolucija robota R_1 i robota R_2 vezani događaji, za *potpuni uspjeh* algoritma potrebno je da se osigura konvergencija oba algoritma, koji nisu vezani. Ako je vjerojatnost konvergencije evolucijskog algoritma označena s A_L, a vjerojatnost konvergencija desnog algoritma s A_D, onda je vjerojatnost potpunog uspjeha:

$$P(A_L \text{ i } A_D) = P(A_L \cap A_D) = P(A_L) \cdot P(A_D), \tag{35}$$

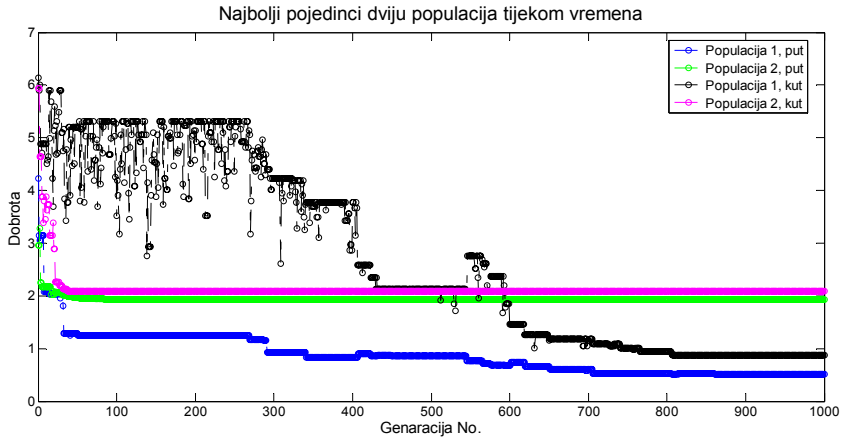
što je, uz uvjet da konvergencija algoritma ne nastupa s vjerojatnošću 1, uvijek manje od vjerojatnosti konvergencija svakog algoritma posebno.

3.2.4 Višekriterijska funkcija dobrote, minimizacija puta i minimizacija kuta

Niti u ovom slučaju nema potrebe za koevolucijom jer je moguće definirati objektivnu funkciju dobrote koja je ovisna za svaku populaciju, o duljini trajektorije što ju opisuje referentna točka na izvršnom članu robota i o ukupnom kutu rotacije u zglobovima robota.



Slika 3-37. Evolucija dva robota u radnom prostoru prema kriterijima minimizacije puta i kuta rotacije.

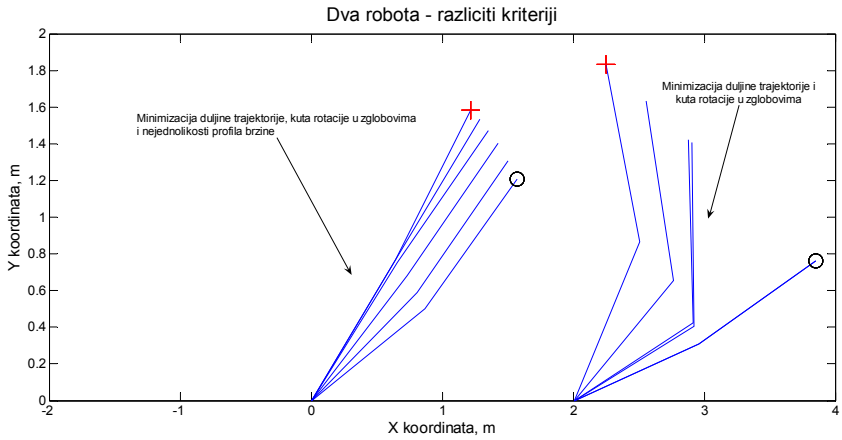


Slika 3-38. Konvergencija populacija prema kriterijima puta i kuta rotacije.

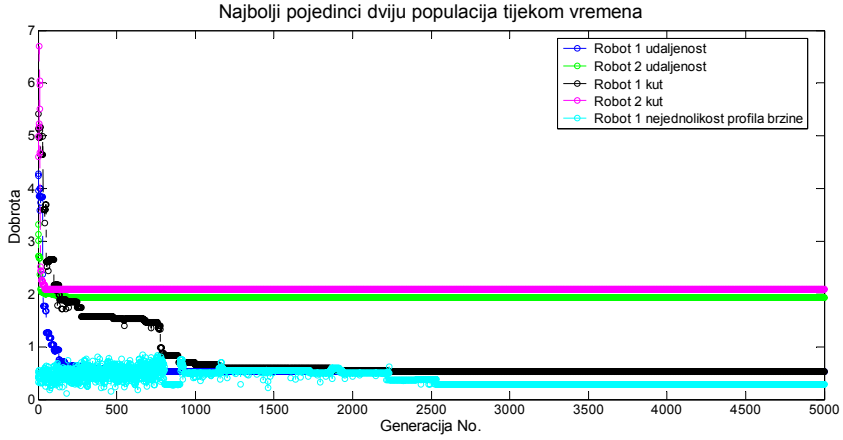
Slike [Slika 3-37] i [Slika 3-38] prikazuju evoluciju pojedinaca koji su vrednovani složenom funkcijom dobrote, definiranom kao linearna kombinacija parcijalnih funkcija dobrote. Nema poteškoća pri konvergenciji, uz moguća odstupanja konačnog rezultata zbog stohastičke prirode algoritma. Asimptotske vrijednosti sa slike [Slika 3-38] razlikuju se, što je posljedica različitih početnih i konačnih konfiguracija robota R_1 i R_2.

3.2.5 Višekriterijska funkcija dobrote; minimizacija puta, minimizacija kuta i minimizacija neravnomjernosti profila brzine

Model je proširen na dvije populacije s tri kriterija – minimizacija duljine trajektorije, minimizacija kuta rotacije u zglobovima i minimizacija nejednolikosti profila brzine referentne točke na izvršnom članu robota.



Slika 3-39. Kod robota R_1 implementirana su tri kriterija, put, kut rotacije i nejednolikost brzine, dok su kod robota R_2, desno, implementirana samo dva kriterija, put i kut.

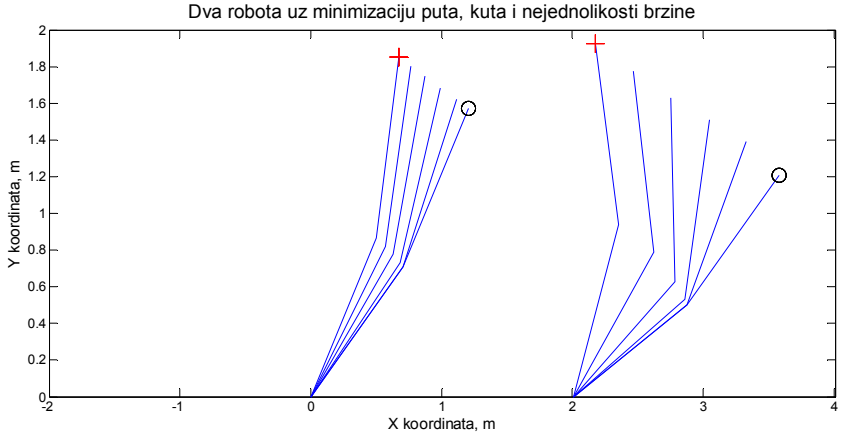


Slika 3-40. Konvergencija populacije za odabrane kriterije.

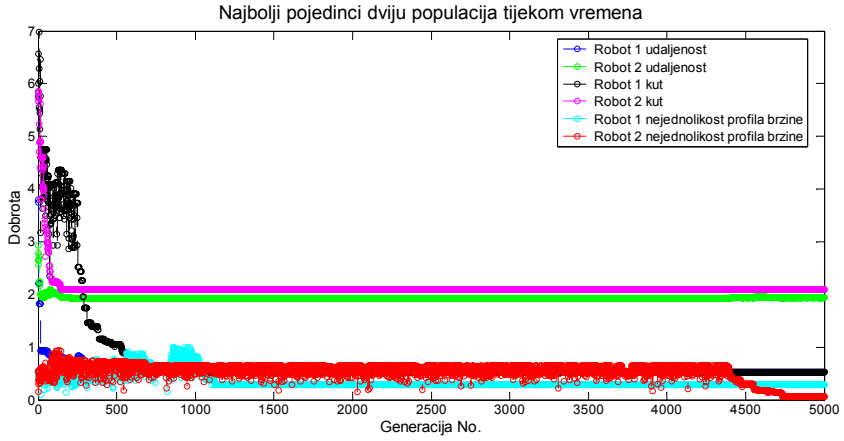
Razlika je između implementacije tri, odnosno samo prva dva kriterija prikazana slikama [Slika 3-39] i [Slika 3-40]. Primjećuje se nejednoliko gibanje robota R_2 (desni robot), odnosno neke konfiguracije prekrivene su inicijalnom ili krajnjom konfiguracijom. U optimalnom slučaju za treći kriterij, sve pozicije u prostoru kroz koje prolazi referentna točka na izvršnom članu robota trebale bi biti jednako udaljene jedna od druge. Uz uvažavanje prethodna dva

kriterija, očekuje se evoluiranje trajektorije koja je najkraća, s minimalnim mogućim promjenama brzine odnosno akceleracije i uz minimalnu utrošenu energiju. Tome bi odgovarala trajektorija lijevoga robota na istoj slici.

Slučaj kada su sva tri kriterija implementirana na oba robota prikazan je na sljedećim slikama [Slika 3-41] i [Slika 3-42]. Oba su robota uspjela evoluirati u prihvatljivo ponašanje, te se približiti u područje blisko globalno-optimalnome. Sljedeći korak je postavljanje robota u takve konfiguracije kod kojih mogu nastupiti kolizije te primjena koevolucijskih koncepata za pretragu prostora stanja. Time se dolazi do takvih ponašanja robota koji, nadograđujući inkrementalnu prilagodbu jedan drugome i uvažavajući prethodno implementirane kriterije evoluiraju kompleksne trajektorije prihvatljive i izvedive na realnome robotskom sustavu



Slika 3-41. Dva robota, tri optimizacijska kriterija, put, kut rotacije i nejednolikost brzine.

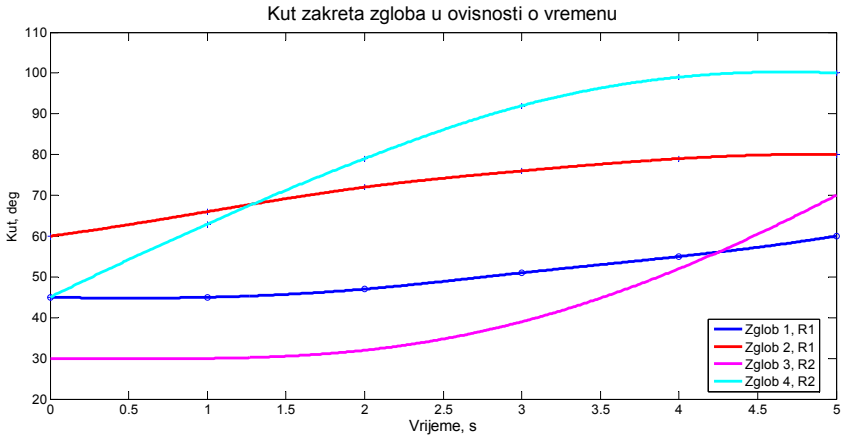


Slika 3-42 Prikaz konvergencije za odabrane kriterije

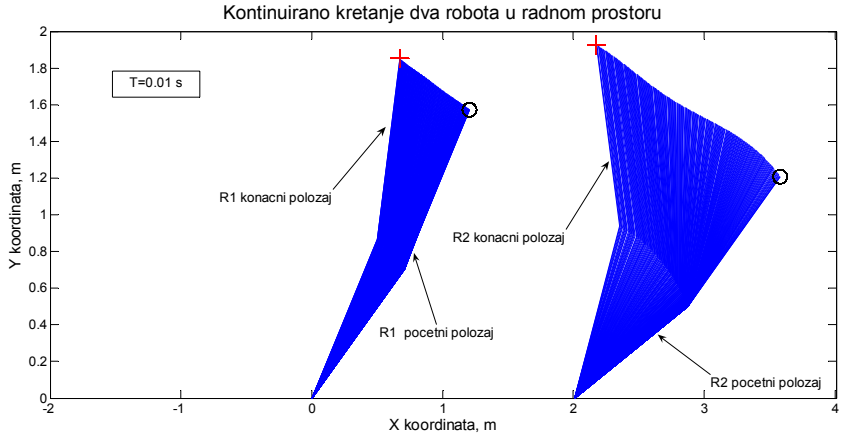
Na slikama [Slika 3-43] i [Slika 3-44] prikazana je kontinuirana promjena kuta zakreta svakog zgloba φ_i u ovisnosti o vremenu te rekonstruirana putanja za robote R_1 i R_2, na temelju rezultata dobivenog evolucijskim algoritmom prikazanog na slici [Slika 3-41]. Rekonstruirana kretanja zadovoljavaju kriterije koji su implementirani i moguće je doseći rješenje koje se nalazi na Pareto fronti. Također, obzirom na vrlo finu diskretizaciju, moguća je implementacija ovako dobivenih rješenja na realne robote 2*R* konfiguracije.

U slučaju kada su početna i konačna konfiguracija agenata definirane tako, da tijekom tranzicije od početnog do konačnog stanja može doći do kolizije između bilo koja dva segmenta robota R_1 i robota R_2, taj slučaj je neprihvatljiv. Bilo koju trajektoriju koja rezultira kolizijom s drugim agentom u proizvoljnom trenutku vremena nije prikladno implementirati na stvarnim robotima. Kod realnog sustava pojava trajektorija bi značila prekid rada sustava, oštećenje robota, a u slučaju kolizije s čovjekom ozljedu. Svaku od ovih mogućnosti potrebno je eliminirati. Sustav dvoručnog robota nije sustav kod kojega je jedan robot upravljiv, dok je drugi statička ili dinamička prepreka. Upravljanjem oba robota moguće je prilagođavati međusobno njihove trajektorije tako da se minimizira broj kolizija, uz istovremenu optimizaciju ostalih prethodno definiranih kriterija.

Proizlazi da su ponašanja ova dva robota međusobno vezana. U tom slučaju, neprikladno je definirati objektivnu mjeru dobrote, koja ne može uzeti u obzir prirodu međudjelovanja dva agenta. Predlaže se primjena koevolucijskog algoritma, pri čemu dobrota svakog agenta iz dviju koevoluirajućih populacija ovisi o tome kako taj agent dobro surađuje s agentima iz druge populacije. Suradnja se u promatranom problemu odnosi na broj kolizija kojima rezultira trajektorija agenta iz prve populacije sa svim agentima (u kanonskom slučaju evaluacije, tzv. potpunog *miješanja* dviju populacija) iz druge populacije. Oni agenti, koji rezultiraju najmanjim brojem kolizija u kombinaciji s agentima iz drugih populacija, primaju veću nagradu na skali dobrote čime se promovira njihovo razmnožavanje i prenošenje informacija o njihovom fenotipu u nadolazeće generacije.



Slika 3-43. Zakret zgloba u ovisnosti o vremenu



Slika 3-44. Kretanje robota sa smanjenim vremenom uzorkovanja, $T = 0.01$ s.

3.2.6 Puni model – dva agenta u radnom prostoru uz početne i konačne uvjete koji rezultiraju kolizijom tijekom gibanja

U slučaju primjene ovakvog koncepta, javljaju se određene specifičnosti u odnosu na standardne evolucijske algoritme. Naime, sada više nije moguće implementirati elitizam, kako je to bilo prikazano u prethodnim poglavljima. Dinamika koevolucije je takva, da dobrota svakog pojedinca iz jedne populacije ovisi o cjelokupnoj distribuciji druge populacije. Obje se populacije mijenjaju, te jedan pojedinac iz prve populacije može u jednom trenutku biti jako dobar, dok već u sljedećem, nakon mutacije, križanja i selekcije druge populacije, njegova dobrota može biti niska. U radu je razvijena metoda čuvanja najboljih *parova* pronađenih tijekom evolucije. Određen postotak najboljih parova čuva se konstantno u posebnoj tablici, te se, ukoliko u trenutnim populacijama nema boljih parova, kopira odgovarajući broj parova iz

tablice koji zamjenjuju najlošije parove trenutne generacije. Time se osigurava čuvanje informacije o najboljem dijelu prostora stanja pronađenog tijekom rada algoritma, čime se *može* ubrzati evolucija. O metodama elitizma govoreno je već ranije, no njihov učinak nije uvijek povoljan, jer u određenim situacijama mogu prouzročiti zastoj pretrage u suboptimalnom području.

Kako je izlaz iz koevolucijskog algoritma diskretan niz konfiguracija, potrebno je provesti razmatranje o među-stanjima, koja nisu dio rješenja dobivenog algoritmom, kako bi se moglo tvrditi da je problem planiranja bez kolizije riješen na cijelom području gibanja oba robota.

3.2.7 Formalna analiza modela dvoručnog robota

Obzirom da je izlaz evolucijskog algoritma diskretan niz konfiguracija robota R_1 i robota R_2 potrebno je pokazati da se između dviju diskretnih konfiguracija može pronaći takva konfiguracija R_1 i R_2 koja osigurava neprekinuto gibanje. Ako je to moguće, onda je diskretizacija gibanja kako je prikazano u ovoj disertaciji prihvatljiva.

Lema 1. *Za planarno gibanje dva robota, R_1 i R_2, definirano početnim i konačnim konfiguracijama u kojima se segmenti robota ne sijeku, postoji takva konfiguracija robota R_1 ili robota R_2 između zadanih početne i konačne konfiguracije, koja osigurava neprekinutu putanju između početne i konačne konfiguracije, klase C_0, tako da su putanje vrhova po dijelovima pravocrtne.*

Uvodi se skup vektora $\{\vec{a},\vec{b},\vec{c},\vec{d}\}$, koji predstavljaju odgovarajuće segmente robota R_1 *i robota* R_2. Svi vektori imaju istu normu, $\|a\| = \|b\| = \|c\| = \|d\| = L$. Kao baza vektorskog prostora odabiru se vektori $\vec{a}$ i $\vec{c}$, za slučaj kada vektori $\vec{a}$ i $\vec{c}$ nisu kolinearni. Vektori $\vec{a}$ i $\vec{c}$ imaju po jednu fiksnu točku, koja predstavlja bazu robota R_1 *i* R_2 i nazivaju se baznim vektorima. Vršni segmenti robota predstavljeni su vektorima $\vec{b}$ i $\vec{d}$, s pripadajućim hvatištima na vrhovima baznih vektora i nazivaju se vršnim vektorima. Vrhovi vršnih vektora određuju referentnu točku robota R_1 ili R_2, kako je prikazano slikom [Slika 3-45].

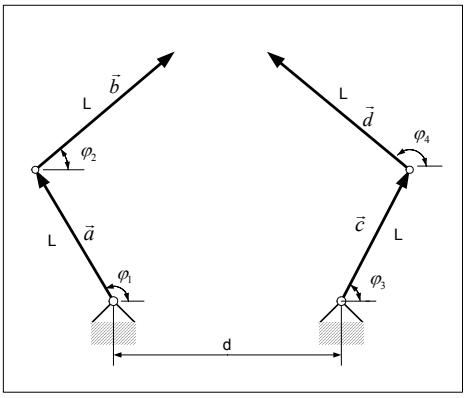
Dokaz: provodi se u tri koraka. U prvom se koraku izvode izrazi za provjeravanje kolizije među segmentima robota izravno pomoću kutova zakreta.

Mogući su sljedeći slučajevi:

Slučaj I.: Sijeku se bazni segmenti, tj. sijeku se vektori $\vec{a}$ i $\vec{c}$.

Slučaj II.: Sijeku se jedan bazni i jedan vršni segment, tj sijeku se vektori $\vec{a}$ i $\vec{d}$ ili vektori $\vec{b}$ i $\vec{c}$.

Slučaj III.: Sijeku se dva vršna segmenta, tj. sijeku se vektori $\vec{b}$ i $\vec{d}$.



Slika 3-45. Definicija dva robota u radnom prostoru.

Uvode se bezdimenzijski parametri μ i ν, o čijim vrijednostima ovisi postojanje presjecišta *P*, pri čemu presjecište *P* postoji ako i samo ako vrijedi $\{\mu,\nu\} \in \langle 0,1]$, gdje su bezdimenzijski parametri definirani na sljedeći način $\overline{O_1P} = \mu\vec{a}$, i $\overline{O_2P} = \nu\vec{c}$.

Parametar $\xi = d/L$ je bezdimenzijska značajka konfiguracije, koju je smisleno promatrati u intervalu: $\xi \in \langle 0,4]$.

Slučaj I.:

Vrijedi sljedeća jednadžba, prema slici [Slika 3-46]:

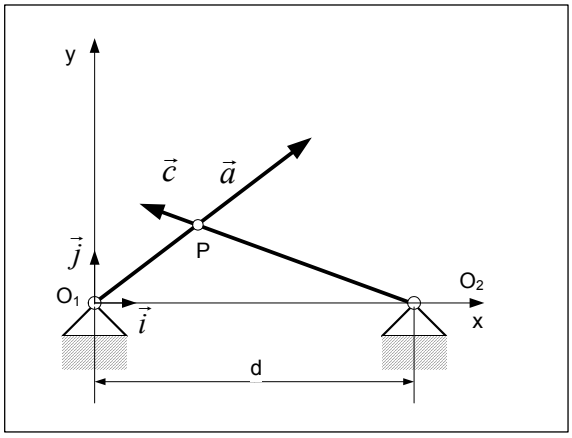
$$\mu\vec{a} = \nu\vec{c} + d\vec{i}. \tag{36}$$

Rješavanjem izraza (36) po nepoznatim parametrima μ i ν dobivaju se sljedeći izrazi za izračun bezdimenzijskih parametara:

$$\mu = \xi \cdot \frac{\sin(\varphi_3)}{\sin(\varphi_3 - \varphi_1)} \text{ i} \tag{37}$$

$$\nu = \xi \cdot \frac{\sin(\varphi_1)}{\sin(\varphi_3 - \varphi_1)}. \tag{38}$$

U slučaju kada je $\varphi_1 = \varphi_3$ bazni su vektori paralelni, presjecište *P* se nalazi u beskonačnosti pa taj slučaj nije potrebno ispitivati. Isto razmatranje primjenjuje se i kod analize slučaja II. i slučaja III.



Slika 3-46. Presjecište dva bazna segmenta.

Slučaj II.:

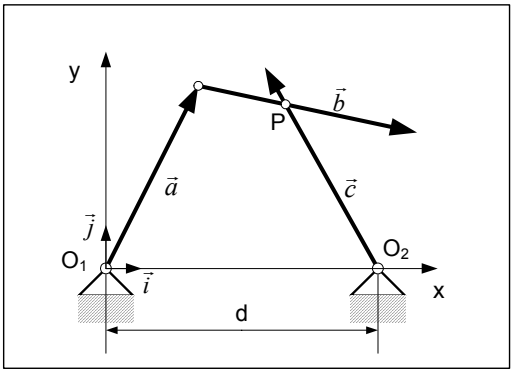
Kada se sijeku jedan vršni vektor, $\vec{b}$ i jedan bazni vektor, $\vec{c}$, može se pisati sljedeći izraz prema slici [Slika 3-47]:

$$\vec{a} + \mu\vec{b} = \nu\vec{c} + d\vec{i}\,. \tag{39}$$

Izrazi za bezdimenzijske parametre su:

$$\mu = \frac{\xi \cdot \sin(\varphi_3) - \sin(\varphi_3 - \varphi_1)}{\sin(\varphi_3 - \varphi_2)}, \tag{40}$$

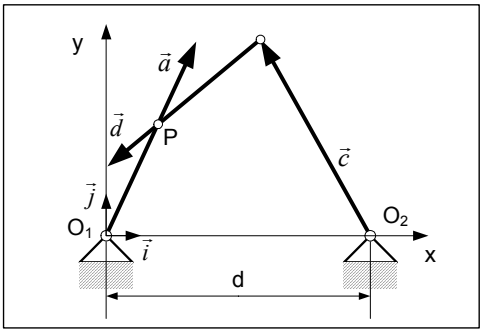
$$\nu = \frac{\xi \cdot \sin(\varphi_2) - \sin(\varphi_2 - \varphi_1)}{\sin(\varphi_3 - \varphi_2)}. \tag{41}$$



Slika 3-47. Slučaj presjecišta vršnog segmenta robota R_1 i baznog segmenta robota R_2.

Analogno, kada se sijeku vršni vektor, $\vec{d}$ i jedan bazni vektor, $\vec{a}$, može se pisati sljedeći izraz prema slici [Slika 3-48]:

$$\mu\vec{a} = \nu\vec{d} + \vec{c} + d\vec{i}. \tag{42}$$



Slika 3-48. Slučaj presjecišta vršnog segmenta robota R_2 i baznog segmenta robota R_1.

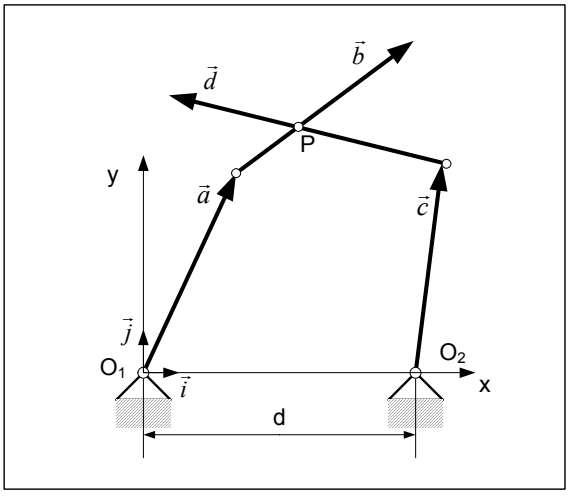
Bezdimenzijski parametri dani su izrazima:

$$\mu = \frac{\xi \cdot \sin(\varphi_1) - \sin(\varphi_1 - \varphi_3)}{\sin(\varphi_1 - \varphi_4)}, \tag{43}$$

$$\nu = \frac{\xi \sin(\varphi_4) - \sin(\varphi_4 - \varphi_3)}{\sin(\varphi_1 - \varphi_4)}. \tag{44}$$

Slučaj III.:

U slučaju presjecišta dva vršna segmenta može se pisati sljedeća jednadžba, prema slici [Slika 3-49]:



Slika 3-49. Presjecište dva vršna segmenta.

$$\vec{a} + \mu\vec{b} = \nu\vec{d} + \vec{c} + d\vec{i}\,. \tag{45}$$

Odgovarajući izrazi za bezdimenzijske parametre su:

$$\mu = \frac{\sin(\varphi_3 - \varphi_4) - \sin(\varphi_1 - \varphi_4) - \xi \cdot \sin(\varphi_4)}{\sin(\varphi_2 - \varphi_4)}, \tag{46}$$

$$\nu = \frac{\sin(\varphi_2 - \varphi_1) + \sin(\varphi_3 - \varphi_2) - \xi \cdot \sin(\varphi_2)}{\sin(\varphi_2 - \varphi_4)}. \tag{47}$$

U drugom koraku razmatra se izlaze evolucijskog algoritma, koji su putanje zadane u *N* diskretnih vremena. Vektori određuju kutove zakreta u zglobovima putem kojih se jednoznačno definiraju konfiguracije robota u diskretnom trenutku *m*. U poglavlju 2.5 pokazano je da su, prema današnjim spoznajama, granice vremenske zahtjevnosti omeđene vremenima $\Theta(n \log n)$ s donje strane i $\Theta(n^n)$ s gornje strane.

Vektor $\vec{\varphi}_{\alpha,m}$ dvokomponentni je vektor s komponentama jednakim kutovima zakreta u odgovarajućem zglobu.

$$\{\vec{\varphi}_{\alpha,m}\} \to \{S^{\beta}_{\alpha,m}\}; \begin{cases} \beta \in \{gornji, donji\} \\ \alpha \in \{R_1, R_2\}, \end{cases} \tag{48}$$

gdje je *S* oznaka segmenta robota α u diskretnom trenutku *m*.

Kolizija između segmenata robota R_1 i robota R_2 definirana je na sljedeći način:

$$I_k(\vec{\varphi}_{\alpha,m}, \vec{\varphi}_{\bar{\alpha},m}) = \begin{cases} = 0 \to nema\ kolizije \\ > 0 \to inače, \end{cases} \tag{49}$$

pri čemu se vrijednost *m* odnosi na diskretan trenutak, $m \in \{1, ..., n\}$. Kolizije se ispituju $\forall m \in \{1, ..., n-2\}$, jer ih prema definiciji problema nije potrebno ispitivati u početnim i konačnim konfiguracijama, a α i $\bar{\alpha}$ oznake su za R_1 i R_2.

Definicija 12. ***Spojnica referentnih točaka:*** Spojnica referentnih točaka na izvršnom članu robota α i $\bar{\alpha}$ u trenucima *i* odnosno *j* definirana je sljedećim izrazom:

$$\Pi_{(\alpha,i,\bar{\alpha},j)} := \begin{pmatrix} \{x,y\} \mid y - y_{\alpha,i} = \dfrac{y_{\bar{\alpha},j} - y_{\alpha,i}}{x_{\bar{\alpha},j} - x_{\alpha,i}}\left(x - x_{\alpha,i}\right), \\ \min\left(x_{\alpha,i}, x_{\bar{\alpha},j}\right) \le x \le \max\left(x_{\alpha,i}, x_{\bar{\alpha},j}\right) \end{pmatrix}, \tag{50}$$

$$\alpha, \bar{\alpha} \in \{R_1, R_2\};\ (i,j) \in \{1,...,n-2\}. \tag{51}$$

Iz definicije 12 slijedi kako je spojnica vrhova jednog robota u dva uzastopna trenutka, m i $m + 1$ dana izrazom:

$$\Pi_{(\alpha,m,\alpha,m+1)} := \begin{pmatrix} \{(x_\alpha, y_\alpha)\} \mid y_\alpha - y_{\alpha,m} = \dfrac{y_{\alpha,m+1} - y_{\alpha,m}}{x_{\alpha,m+1} - x_{\alpha,m}}\left(x_\alpha - x_{\alpha,m}\right), \\ \min\left(x_{\alpha,m}, x_{\alpha,m+1}\right) \le x \le \max\left(x_{\alpha,m}, x_{\alpha,m+1}\right) \end{pmatrix}. \tag{52}$$

Definicija 13. ***Presjecište segmenata robota i spojnice vrhova robota.*** Presjecište je definirano kako slijedi:

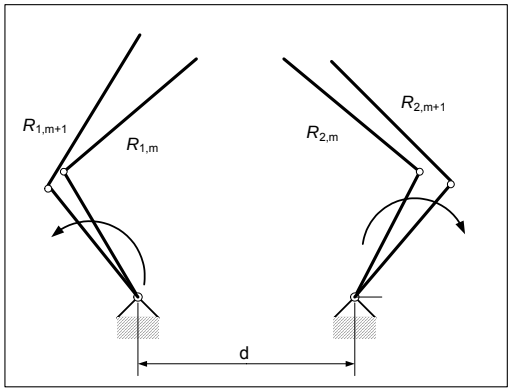
$$\begin{aligned} &(x^*, y^*) \mid (x^*, y^*) \in \Pi_{\alpha,m,\alpha,m+1} \wedge (x^*, y^*) \in S^{\beta}_{\bar{\alpha},m} \cup S^{\bar{\beta}}_{\bar{\alpha},m}; \\ &(\alpha, \bar{\alpha}) \in \{R_1, R_2\} \mid \alpha \neq \bar{\alpha};\ (\beta, \bar{\beta}) \in \{donji, gornji\} \mid \beta \neq \bar{\beta}, \end{aligned} \tag{53}$$

odnosno, skup $\mathcal{PR}^*$, koji obuhvaća sva presjecišta (x^*, y^*) definiran je sljedećim izrazom:

$$\mathcal{PR}^*_{\alpha,m} := \Pi_{\alpha,m,\alpha,m+1} \cap \left(S^{\beta}_{\bar{\alpha},m} \cup S^{\bar{\beta}}_{\bar{\alpha},m}\right). \tag{54}$$

U trećem koraku se pokazuje kako osigurati kontinuiranu putanju robota između dva uzastopna trenutka. Za dvije putanje generirane algoritmom ispituju se sljedeća tri slučaja:

Slučaj A: Površina koju zatvaraju dvije uzastopne konfiguracije robota R_1 i robota R_2 nemaju dodirnih točaka, slika [Slika 3-50]:



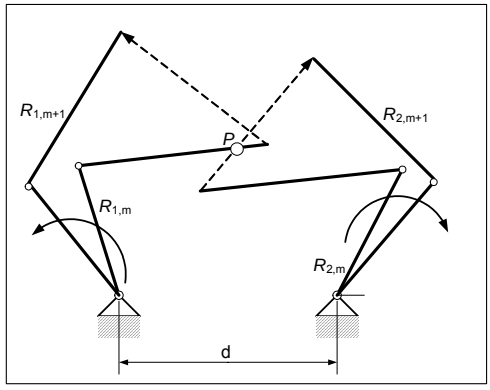
Slika 3-50. Površine robota bez dodirnih točaka.

U ovom slučaju presjecište ne postoji, vrijedi:

$$P_{\alpha,m} \cup P_{\overline{\alpha},m} = \varnothing. \tag{55}$$

Ovo je slučaj kada se roboti mogu gibati bez ograničenja, postoji rješenje primjenom evolucijskog algoritma, kako je prikazano poglavljem 3.2.5.

Slučaj B: Postoji točno jedno presjecište, *P*, spojnice vrhova jednog robota i početne i konačne konfiguracije drugog robota:



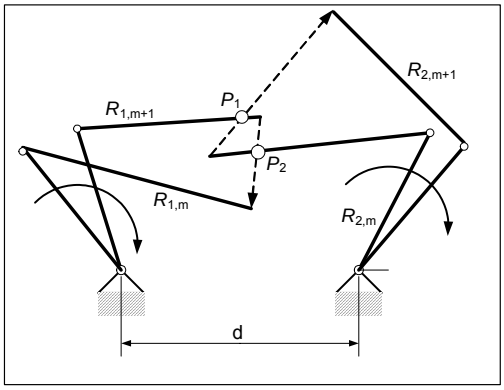
Slika 3-51. Jedno presjecište.

$$\exists \alpha \mid P_{\alpha,m} = \varnothing \wedge P_{\overline{\alpha},m} \neq \varnothing. \tag{56}$$

Rješenje za gibanje bez kolizija postoji u ovom slučaju i to tako, da se prvo giba robot α, a potom robot $\overline{\alpha}$.

Slučaj C: Postoje dva presjecišta, $P_{\alpha,m} \neq \varnothing \neq P_{\overline{\alpha},m}$. Ovo je općeniti slučaj, kod kojega je potrebno konstruirati putanju robota koja rezultira neprekinutim gibanjem robota R_1 i

robota R_2, bez kolizije segmenata robota te bez presijecanja trajektorije bilo kojeg robota sa segmentom drugog robota, slika [Slika 3-52].



Slika 3-52. Opći slučaj, potrebno je konstruirati putanju.

Duljina spojnice dvije referentne točke na izvršnom članu robota R_1 i R_2 dana je sljedećim izrazom:

$$\begin{aligned} &s = \max\ d\left[\left(x_1, y_1\right); \left(x_2, y_2\right)\right]; \\ &\left(x_1, y_1\right) \in \Pi\left(\alpha, m, \overline{\alpha}, m\right), \\ &\left(x_2, y_2\right) \in \Pi\left(\alpha, m, \overline{\alpha}, m\right). \end{aligned} \tag{57}$$

Funkcija $d(T_1,T_2)$ daje euklidsku udaljenost točaka T_1 i T_2.

Kut što ga spojnica dviju referentnih točaka zatvara s osi koja prolazi kroz fiksne točke robota R_1 i R_2 u nekom trenutku m dana je sljedećim izrazom:

$$\theta\left(\alpha, m; \overline{\alpha}, m\right) := \frac{y_{\overline{\alpha},m} - y_{\alpha,m}}{x_{\overline{\alpha},m} - x_{\alpha,m}}. \tag{58}$$

Kut spojnice u početnom trenutku, trenutak m:

$$\theta\left(\alpha, m; \overline{\alpha}, m\right) \tag{59}$$

i u konačnom trenutku, trenutak $m + 1$:

$$\theta\left(\alpha, m+1; \overline{\alpha}, m+1\right). \tag{60}$$

Težište spojnice dviju referentnih točaka u trenutku m definirano je sljedećim izrazom:

$$T\left(\alpha, m; \overline{\alpha}, m\right) = \left(x_T, y_T\right) = \left(\frac{x_{\alpha,m} + x_{\overline{\alpha},m}}{2}, \frac{y_{\alpha,m} + y_{\overline{\alpha},m}}{2}\right). \tag{61}$$

Vektor koji opisuje položaj težišta spojnice u trenutku *m* definiran je izrazom:

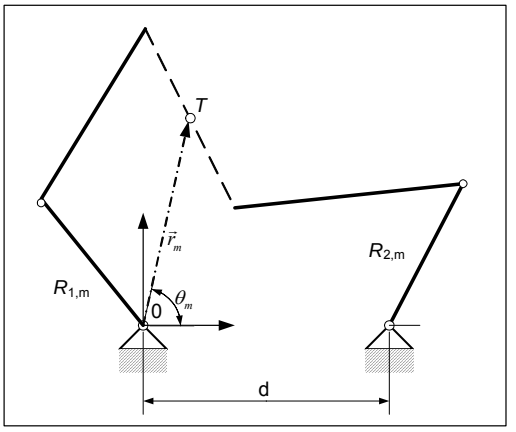
$$\vec{r} := \overrightarrow{OT};\ O = (0,0);\ \vec{r}_m := \overrightarrow{OT}(\alpha, m, \bar{\alpha}, m). \tag{62}$$

Putanja robota kojom se izbjegava kolizija bit će opisana kao promjena uvedenih parametara:

$$\left(s_m, \theta(\alpha, m; \bar{\alpha}, m), \vec{t}_m\right) \to \left(s_{m+1}, \theta(\alpha, m+1; \bar{\alpha}, m+1), \vec{t}_{m+1}\right), \tag{63}$$

pri čemu se θ mijenja strogo monotono između vrijednosti na početku i na kraju promatranog intervala:

$$\theta \mid \theta(\alpha, m; \bar{\alpha}, m) \to \theta(\alpha, m+1; \bar{\alpha}, m+1). \tag{64}$$



Slika 3-53. Definicija težišta spojnice i vektora položaja.

Vektor težišta mijenja se prema izrazu:

$$\vec{r}_t - \vec{r}_m = (\vec{r}_{m+1} - \vec{r}_m)\frac{\theta_t - \theta(\alpha, m; \bar{\alpha}, m)}{\theta(\alpha, m+1; \bar{\alpha}, m+1) - \theta(\alpha, m; \bar{\alpha}, m)}, \tag{65}$$

pri čemu je θ_t prikloni kut spojnice u trenutku $t \mid t_m \le t \le t_{m+1}$.

Spojnica referentnih točaka mijena se prema izrazu:

$$s_t = s_m + (s_{m+1} - s_m)\frac{\theta_t - \theta(\alpha, m; \bar{\alpha}, m)}{\theta(\alpha, m+1; \bar{\alpha}, m+1) - \theta(\alpha, m; \bar{\alpha}, m)}. \tag{66}$$

Y koordinata težišta *T* dana je sljedećim izrazom:

$$\vec{r}_t \cdot \vec{j} = \left[\vec{t}_m + \left(\vec{t}_{m+1} - \vec{t}_m \right) \frac{\theta_t - \theta(\alpha, m; \bar{\alpha}, m)}{\theta(\alpha, m+1; \bar{\alpha}, m+1) - \theta(\alpha, m; \bar{\alpha}, m)} \right] \cdot \vec{j} =$$
$$= \frac{y_{\alpha,m} + y_{\bar{\alpha},m}}{2} + \frac{y_{\alpha,m+1} + y_{\bar{\alpha},m+1} - \left(y_{\alpha,m} + y_{\bar{\alpha},m} \right)}{2} \cdot \frac{\theta_t - \theta(\alpha, m; \bar{\alpha}, m)}{\theta(\alpha, m+1; \bar{\alpha}, m+1) - \theta(\alpha, m; \bar{\alpha}, m)}, \tag{67}$$

$$y_{\alpha,t} = \frac{y_{\alpha,m} + y_{\bar{\alpha},m}}{2} + \frac{y_{\alpha,m+1} + y_{\bar{\alpha},m+1} - \left(y_{\alpha,m} + y_{\bar{\alpha},m} \right)}{2} \cdot \frac{\theta_t - \theta(\alpha, m; \bar{\alpha}, m)}{\theta(\alpha, m+1; \bar{\alpha}, m+1) - \theta(\alpha, m; \bar{\alpha}, m)} \pm \frac{s_t}{2} \sin \theta_t$$

Predznak e određuje iz početnog uvjeta:

$$y_{\alpha,m} = \frac{y_{\alpha,m} + y_{\bar{\alpha},m}}{2} \pm \frac{s_t}{2} \sin \theta(\alpha, m; \bar{\alpha}, m) = \frac{y_{\alpha,m} + y_{\bar{\alpha},m}}{2} \pm \frac{y_{\alpha,m} - y_{\bar{\alpha},m}}{2}, \tag{68}$$

bira se gornji predznak.

Preostaje izraziti opisanu putanju u polaznim varijablama; $\vec{\varphi}$, za što se prvo izražavaju koordinate referentnih točaka izvršnih članova robota. Za kut što ga spojnica vrhova robota R_1 i R_2 zatvara s osi koja prolazi kroz fiksne točke vrijedi:

$$\theta(\alpha, m; \bar{\alpha}, m) : \operatorname{tg} \theta(\alpha, m; \bar{\alpha}, m) = \frac{y_{\bar{\alpha},m} - y_{\alpha,m}}{x_{\bar{\alpha},m} - x_{\alpha,m}}. \tag{69}$$

Vektor težišta može se u trenutku *m* zapisati preko koordinata referentnih točaka pomoću izraza:

$$\vec{r}_m = \frac{x_{\alpha,m} + x_{\bar{\alpha},m}}{2} \vec{i} + \frac{y_{\alpha,m} + y_{\bar{\alpha},m}}{2} \vec{j}. \tag{70}$$

Za spojnicu referentnih točaka vrijedi:

$$s_m = \frac{\left(y_{\alpha,m} - y_{\bar{\alpha},m} \right)}{\left| \sin \theta_m \right|}. \tag{71}$$

Vrijede sljedeće relacije za referentne točke robota R_1 i robota R_2 u trenutku *t* i za težište spojnice *s*:

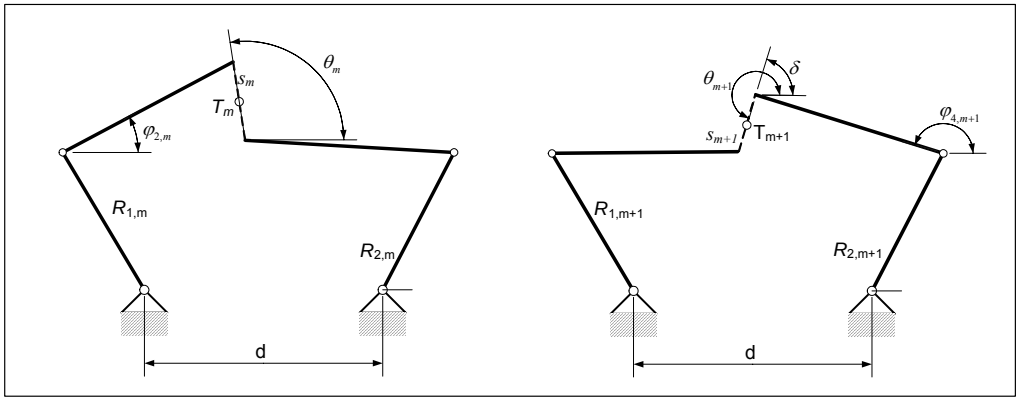
$$\begin{aligned} x_{\alpha t} + x_{\bar{\alpha} t} &= 2\vec{r}_t \cdot \vec{i} \\ x_{\alpha t} - x_{\bar{\alpha} t} &= \pm s_t \cos \theta_t \\ y_{\alpha t} + y_{\bar{\alpha} t} &= 2\vec{r}_t \cdot \vec{j} \\ y_{\alpha t} - y_{\bar{\alpha} t} &= \pm s_t \sin \theta_t, \end{aligned} \tag{72}$$

čije rješavanje daje:

$$
\begin{aligned}
x_{\alpha t} &= \vec{r}_t \cdot \vec{i} \pm \frac{s_t}{2}\cos\theta_t \\
x_{\bar{\alpha} t} &= \vec{r}_t \cdot \vec{i} \mp \frac{s_t}{2}\cos\theta_t \\
y_{\alpha t} &= \vec{r}_t \cdot \vec{j} \pm \frac{s_t}{2}\sin\theta_t \\
y_{\bar{\alpha} t} &= \vec{r}_i \cdot \vec{j} \mp \frac{s_t}{2}\sin\theta_t .
\end{aligned}
\tag{73}
$$

Budući da je spojnica vrhova robota s_t strogo monotona funkcija vremena, određena linearnom funkcijom kuta θ, te budući da su krajnje (početna i konačna) dužina prema definiciji različite od nule, slijedi da niti u jednom trenutku ne dolazi do dodira vrhova.

Postavlja se dodatni uvjet na način rotacije spojnice kako bi se izbjegla mogućnost dodira vrha jednog robota sa segmentom drugog robota. U razmatranju će se uzeti da vrijedi sljedeći odnos kutova u početnom trenutku: $\theta_m > \varphi_{2,m}$, kako je prikazano slikom [Slika 3-54].

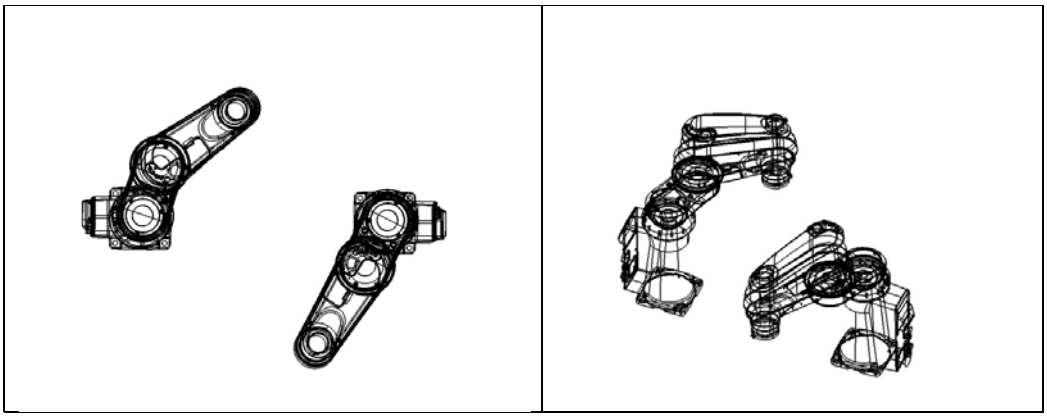


Slika 3-54. Rotacija spojnice *s* oko težišta *T*.

Rotacija spojnice s izvodi se oko težišta spojnice T. Dodir referentne točke robota R1 s vršnim segmentom robota R2 nastupa kada je $\theta = \pi + \varphi_4$. Slijedi da uz uvjet monotonosti kuta θ i $\theta < \pi + \varphi_4$ ne dolazi do dodira vršnih segmenata robota. Uz uvjet $\xi > 2$, što je kod realnih sustava u velikoj većini slučajeva ispunjeno, nije potrebno konstruirati putanju s obzirom na bazne segmente. U suprotnom, lako se pokazuje da izneseno razmatranje vrijedi i za bazne segmente.■

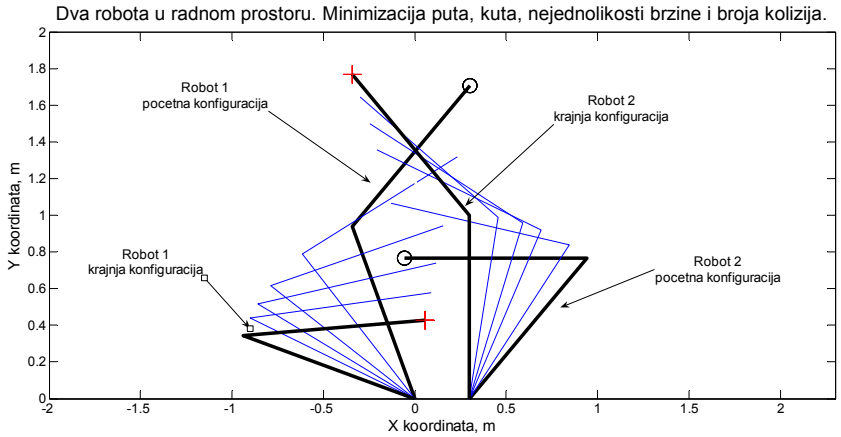
3.2.8 Rezultati simulacija za dva agenta uz različite početne i konačne uvjete

Obzirom na razmatranje provedeno u prethodnom poglavlju, za analizu je zanimljiv slučaj C, kada roboti zauzimaju takav položaj da ne postoji način prelaska iz početne u konačnu konfiguraciju osim međusobnog usklađivanja kretanja, slučaj C je prikazan slikom [Slika 3-55] za dva robota 2R konfiguracija.



Slika 3-55. Dva SCARA robota u radnom prostoru.

Izabrana je konfiguracija prikazana slikom [Slika 3-56] za koju je potrebno pronaći putanje koje ne rezultiraju kolizijama za dva agenta u radnom prostoru.

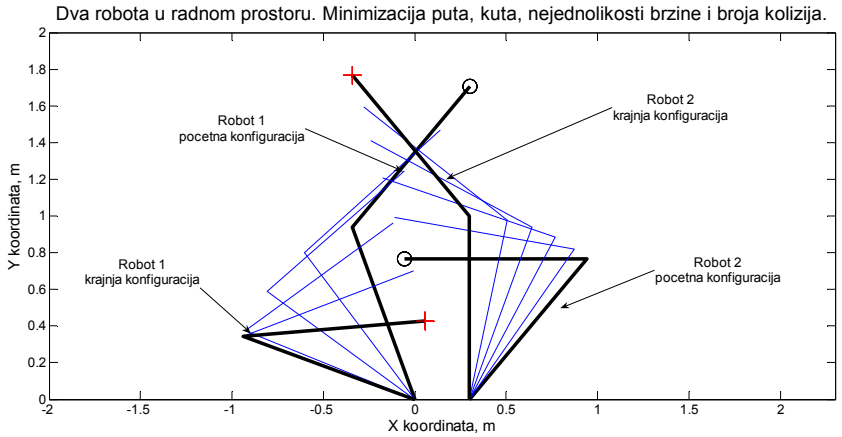


Slika 3-56. Slučaj s dva agenta, vrednovanje kolizije nije uključeno u izračun dobrote.

Iako je implementiran koevolucijski mehanizam za evaluaciju kolizija, još nije uključen u sam izračun funkcije dobrote, kako bi se ilustrirao slučaj kada se roboti gibaju ne

uzimajući u obzir informacije o drugom agentu. Primjećuje se da će doći do kolizije prilikom tranzicije iz druge u treću konfiguraciju agenata.

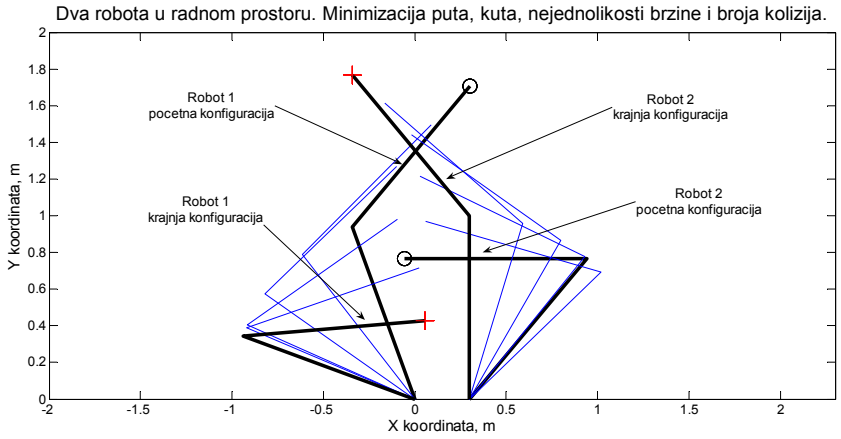
Sljedeća slika pokazuje, uz početne i konačne uvjete iste prethodnima, razvijeno ponašanje nakon što se na robot R_1 primijeni vrednovanje kolizija, dok robot R_2 ostaje isti kao u prethodnom slučaju.



Slika 3-57. Slučaj s dva agenta, vrednovanje kolizija uključeno na agentu R_1.

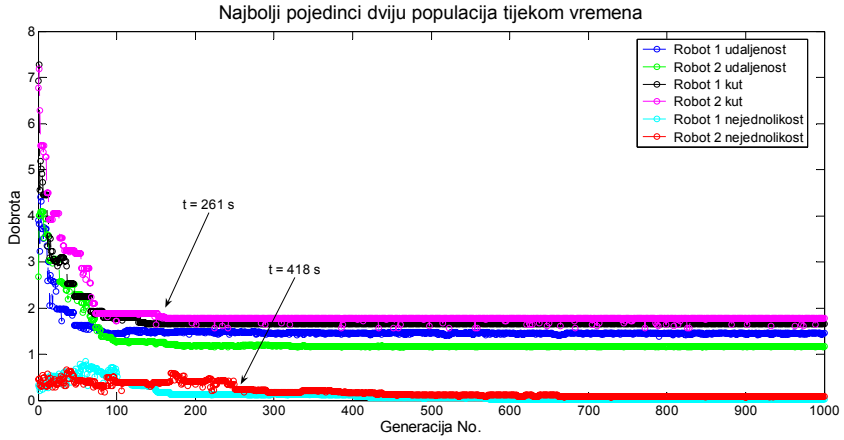
U ovom slučaju, vidi se odmicanje robota R_1 iz zone mogućeg sudara, ali u zadanom broju iteracija još nije pronađeno zadovoljavajuće rješenje, koje rezultira neprekinutim gibanjem bez kolizija.

Slika [Slika 3-58] prikazuje uspješno rješenje problema planiranja putanje. Razvijene su takve trajektorije koje ni u jednom trenutku ne rezultiraju kolizijama, kako u diskretnim vremenskim intervalima, tako niti u interpoliranim međufazama. Agenti međusobno prilagođavaju kretanje na takav način ad se odmiču iz zone u kojoj postoji mogućnost sudara.



Slika 3-58. Uspješno rješenje planiranja putanje za dva agenta u radnom prostoru.

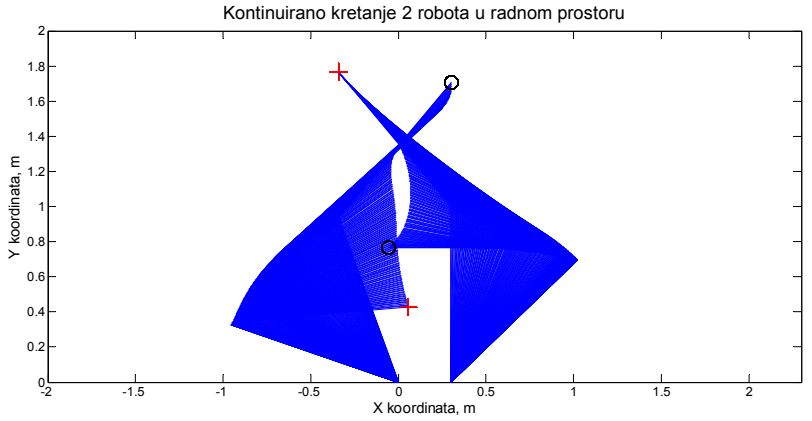
Pripadajući dijagram konvergencije prikazan je slikom [Slika 3-59]. Odgovarajuća vremena istaknuta su na slici, nakon kojih je napredak zanemarivo malen.



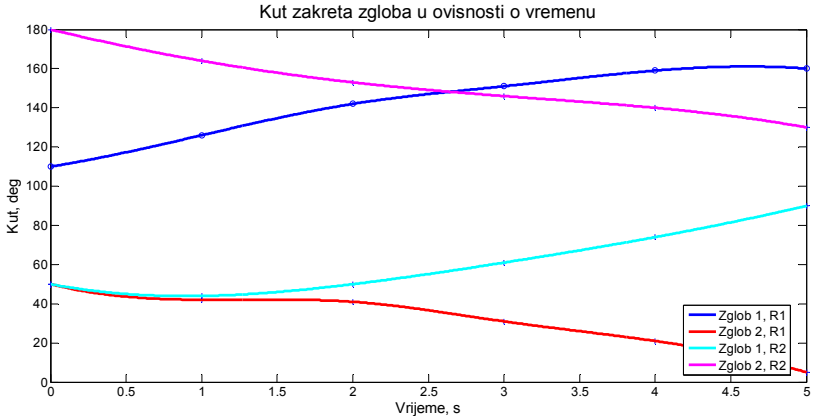
Slika 3-59. Prikaz konvergencije najboljih pojedinaca iz dviju populacija.

Iako prethodno rješenje rješava problem pronalaženja trajektorije bez kolizije, ostaje mogućnost daljnjeg poboljšavanja prvenstveno kriterija nejednolikosti profila brzine. Kontinuirani prikaz kretanja dva robota s odabranim početnim i konačnim konfiguracijama prikazan je slikom [Slika 3-60]. Primjećuje se da se agenti prilagođavaju uzajamno kako bi se izbjegla pojava kolizija u radnome prostoru. Kod ovisnosti zakreta zglobova o vremenu ne pojavljuju se izraženi gradijenti, [Slika 3-61] te su razvijene rješenja pogodna za implementaciju na sustavu sastavljenom od dva realna robota. Odabrano diskretizacijsko vrijeme je 0.01 s, uz mogućnost daljnjeg profinjavanja. Bitno je da je sada inkrementalni kut

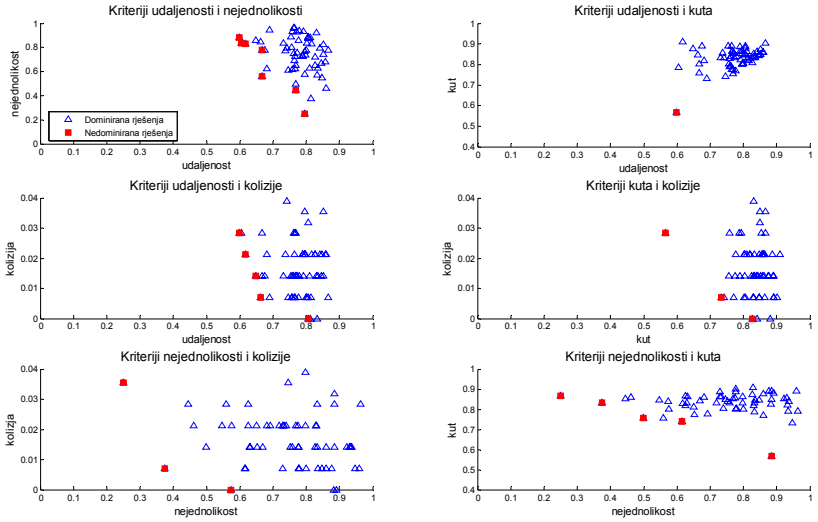
za koji se robot pomiče između dviju uzastopnih konfiguracija znatno manji od teoretski izračunatog maksimalnog dopuštenog kuta, poglavlje 3.1.



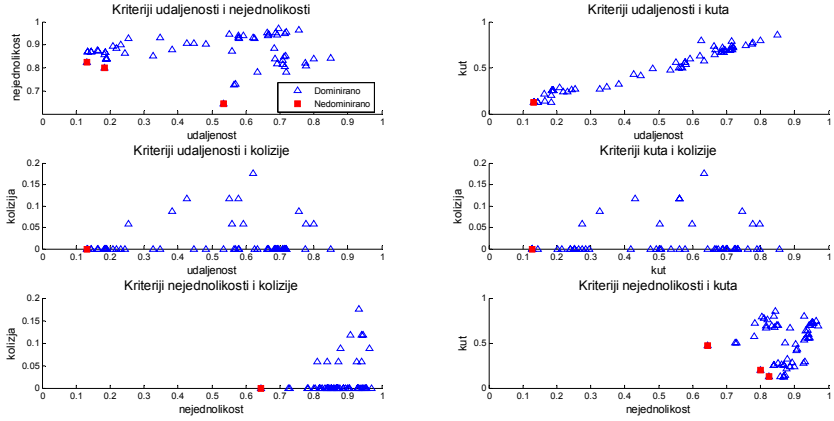
Slika 3-60. Kontinuirano kretanje uz prilagođavanje kretanja radi izbjegavanja kolizija.



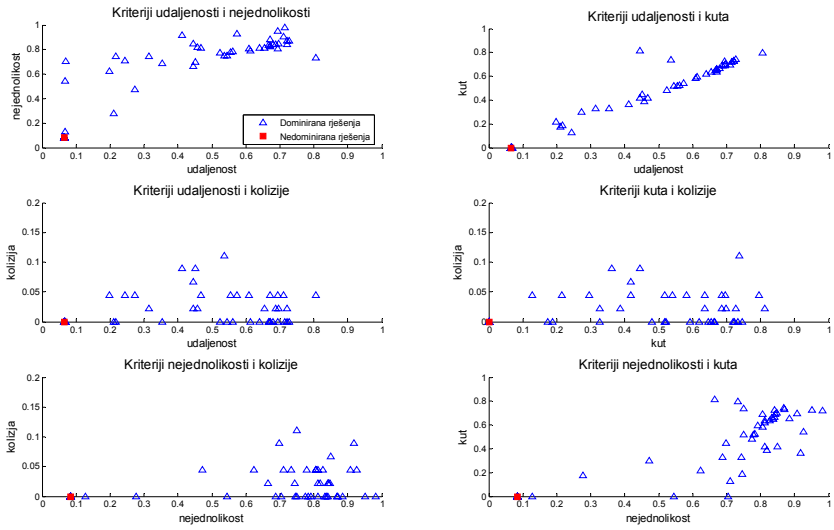
Slika 3-61. Prikaz promjene zakreta kuta u zglobovima robota R_1 i robota R_2.


Kriteriji udaljenosti i nejednolikosti
nejednolikost
udaljenost
Dominirana rješenja
Nedominirana rješenja
Kriteriji udaljenosti i kuta
kut
udaljenost
Kriteriji udaljenosti i kolizije
kolizija
udaljenost
Kriteriji kuta i kolizije
kolizija
kut
Kriteriji nejednolikosti i kolizije
kolizija
nejednolikost
Kriteriji nejednolikosti i kuta
kut
nejednolikost

a)


Kriteriji udaljenosti i nejednolikosti
nejednolikost
udaljenost
Dominirano
Nedominirano
Kriteriji udaljenosti i kuta
kut
udaljenost
Kriteriji udaljenosti i kolizije
kolizija
udaljenost
Kriteriji kuta i kolizije
kolizija
kut
Kriteriji nejednolikosti i kolizije
kolizija
nejednolikost
Kriteriji nejednolikosti i kuta
kut
nejednolikost

b)

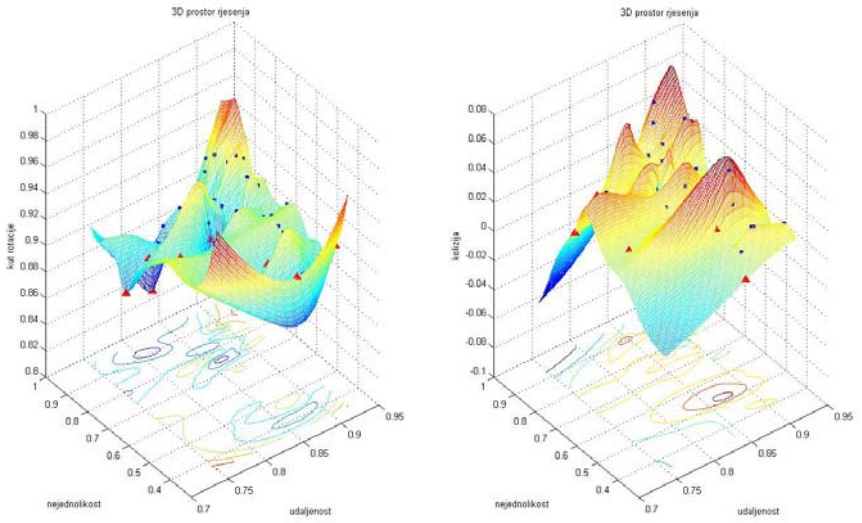


c)

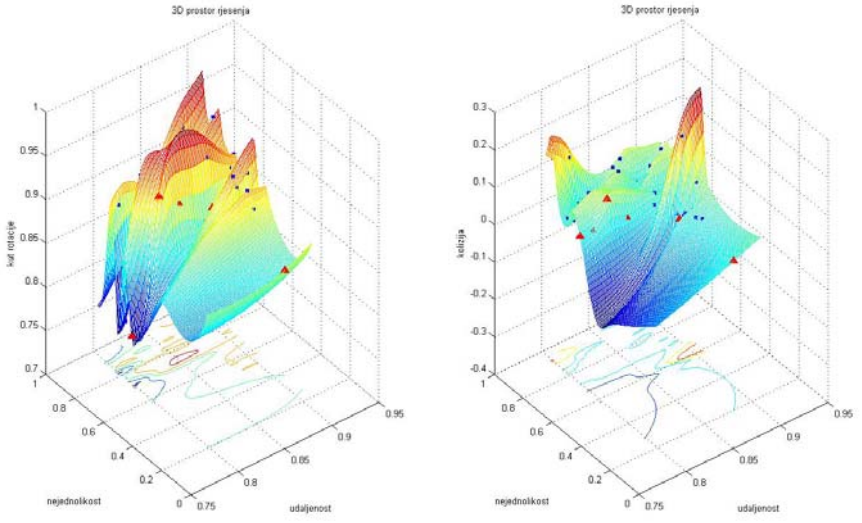
Slika 3-62. Prikaz nedominiranih rješenja za agenta R_1 u radnom prostoru tijekom evolucije; a) G = 10 generacija; b) G = 100 generacija i c) G = 400 generacija – rješenje prikazano slikom [Slika 3-60]. Crveno – nedominirana rješenja; plavo – dominirana rješenja. Skale vrijednosti su normirane u području [0,1]. Veličina populacije N = 60; duljina kromosoma l = 8.

Slikom [Slika 3-62] prikazana je evolucija rješenja dobivenih evolucijskim algoritmom, uz implementiranu metodu selekcije temeljenu na Pareto dominaciji, što je objašnjeno u poglavlju 2.7. Plavom bojom su prikazana dominirana rješenja u trenutnoj populaciji, dok su crvenom bojom prikazana nedominirana rješenja pronađena evolucijskim algoritmom. Primjećuje se konvergencija prema točki najboljih karakteristika – položaj gdje su minimizirani svi kriteriji optimizacije. Također je vidljivo da kriterij optimizacije duljine trajektorije ne postiže optimalnu vrijednost, što je posljedica činjenice da se robot mora odmaknuti od idealne putanje, iako je tijekom evolucije zabilježeno jedno nedominirano rješenje s normiranom duljinom jednakom 0 – slika b). Kod kriterija nejednolikosti još ima prostora za napredak, odnosno nije dosegnut globalni optimum tog kriterija koji je jednak 0. Iako se tijekom evolucije primjećuje zgušnjavanje oblaka točaka koji predstavljaju potencijalna rješenja problema, i kod generacije 400 još je uvijek prisutna određena raznolikost u populaciji. Također je primjetno različito ponašanje oblaka točaka za diskretne kriterije – poput broja kolizije pojedinca u odnosu na broj članova populacije i funkcija koje imaju kontinuiranu karakteristiku – poput kuta zakreta u zglobovima. Primjenom ovog koncepta dominacije rješenja može se utjecati na tijek izvođenja algoritma, odnosno na

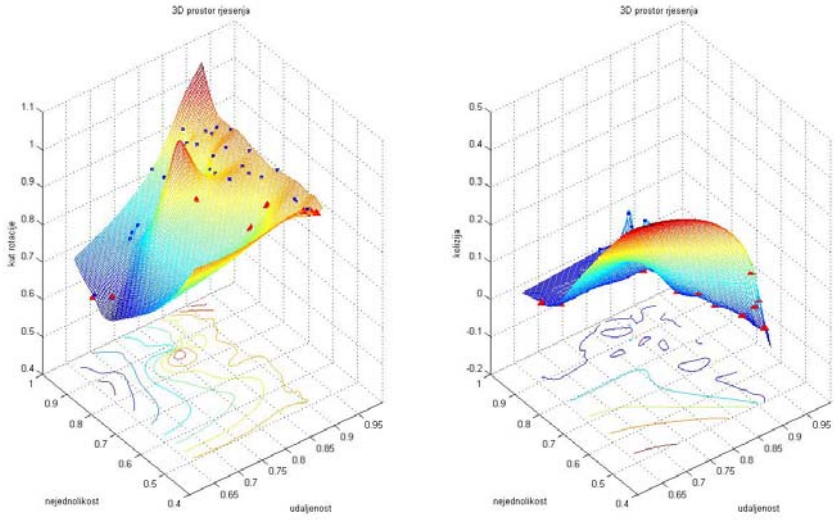
njegov završetak, te se, kopirajući ova rješenja u nadolazeće populacije može osigurati brža konvergencija algoritma prema globalno – optimalnom području.



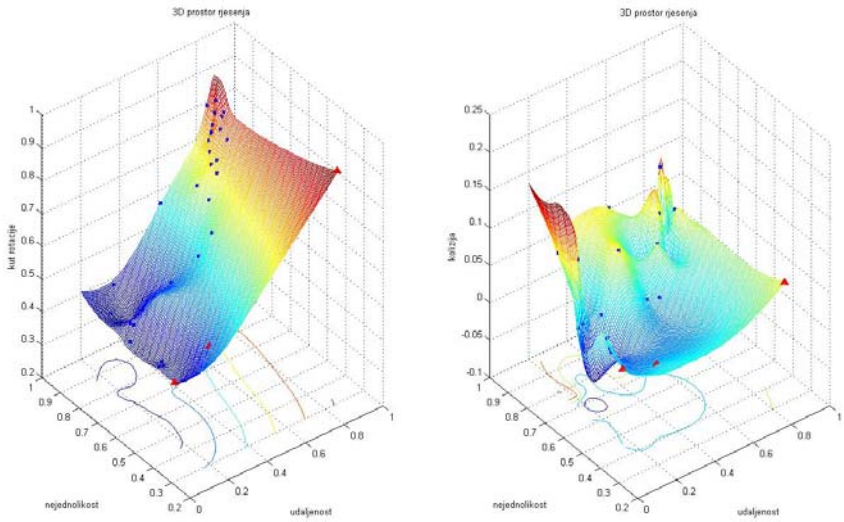
a)



b)



c)



d)

Slika 3-63. Prostorni prikaz konvergencije. Slika a) $G = 1$; b) $G = 50$; c) $G = 200$ i d) $G = 400$. Lijeva strana slike pokazuje kombinaciju kriterija nejednolikosti, udaljenosti i kuta rotacije, desna strana slike pokazuje kombinaciju kriterija nejednolikosti, udaljenosti i broja kolizija.

Slikom [Slika 3-63] a) – d) prikazan je promjena prostora stanja za populaciju rješenja kroz referentne generacije, od početne, do konačne s pronađenim optimalnim rješenjem višekriterijskog problema. Kod proučavanja višekriterijskog optimiranja primjenom evolucijskih algoritama, situacija je gotovo uvijek takva da se algoritmi testiraju na

odgovarajućem skupu referentnih tzv. *benchmark* problema, poput Ackleyeve, Schwefelove ili poopćene Rastriginove funkcije za multimodalne domene probleme. U tom slučaju, prostori prikazani gornjom slikom poznati su unaprijed što omogućuje komparaciju predloženih algoritama. Prostor rješenja kod većine stvarnih problema nije poznat unaprijed, što je i slučaj kod problema koji se rješava u ovoj disertaciji. Ispitivanjem ovog prostora, i uočavanjem njegovih značajki, bilo bi moguće implementirati izvornu, nestandardnu metodu inicijalizacije početnih rješenja. Uobičajeno se početna populacija inicijalizira po slučaju, kako je već ranije objašnjeno. Na slici [Slika 3-63] svakom rješenju koje je prikazano plavim pravokutnikom, za slučaj dominiranog i crvenim trokutom, za slučaj nedominiranog rješenja, pripada odgovarajući vektor kutova. Taj vektor, u slučaju nedominiranog rješenja, predstavlja najbolje pronađeno rješenje unutar populacije za dani broj generacija.

Ova informacija može se iskoristiti na sljedeći način: u slučaju *sličnog* problema koji se javlja u budućem radu robota, npr. kada dva robota imaju početne i konačne konfiguracije slične već pronađenom poznatom rješenju, u idealnom slučaju, može se pozvati to poznato rješenje koje je već spremljeno u generiranoj bazi podataka. U tom slučaju, nije niti potrebno izvoditi pretragu prostora stanja. Zanimljiviji je slučaj kada su početni i konačni uvjeti *donekle* slični, te direktno pozivanje poznatog rješenja ne rješava promatrani problem na zadovoljavajući način. Tada je moguće prilikom inicijalizacije populacije kopirati određeni broj poznatih, „dobrih" rješenja, koja kao nukleusi evoluciju onda ubrzano vode u željenom smjeru i rezultiraju kraćim vremenom potrebnim za konvergenciju.

Ove su dvije metode implementirane u našim prethodnim radovima, ali umjesto baze znanja korištena je samoorganizirajuća neuronska mreža ART1[10] tipa [73] [74]. Osnovna motivacija je bila razvoj agenta sposobnog za učenje. Agent počinje kao *tabula rasa* i tijekom vremena, njegove se sposobnosti povećavaju. Pokazan je koristan učinak implementacije ekspertnog znanja u inicijalizaciju populacija.

3.3 Prilagodba parametara koevolucijskog algoritma

Na pitanje poput „Koja je optimalna vjerojatnost mutacije?" za određeni evolucijski algoritam nema odgovarajućeg odgovora. Jedna od osnovnih poteškoća kod oblikovanja i kasnije primjene evolucijskog algoritma je u nelinearnim i teško dokučivim vezama između pojedinih

[10] ART – eng. *Adaptive Resonance Theory* neuronska mreža.

parametara koji određuju rad evolucijskog algoritma. Zato se često kaže da je jedan problem napraviti evolucijski algoritam koji rješava neki zadatak, dok je drugi problem ugoditi parametre algoritma da bi rješavanje zadaće bilo efikasno. Drugim riječima, u osnovi istim evolucijskim algoritmom mogu se dosizati različite razine uspješnosti na istom problemu.

Nužan uvjet za uspjeh evolucijskog algoritma na određenom problemu je konstruiranje odgovarajuće funkcije dobrote. Pravila za to kako će specifična funkcija dobrote izgledati ne postoje, no može se reći da bi trebala uključivati minimalan broj parametara kojima je moguće na zadovoljavajući način evaluirati pojedince i konačno, razviti ih od početnog stanja niske dobrote do konačnog stanja gdje postižu visoke razine dobrote. Funkcija dobrote slika je okoline kojoj se pojedinci moraju prilagoditi i ima samim time centralnu ulogu u radu bilo kojeg evolucijskog algoritma. Na funkciji dobrote temelji se postupak selekcije, koji je zaslužan za promoviranje onih jedinki koje imaju veću dobrotu u sljedeće populacije.

Mutacija, implementirana u evolucijske algoritme kao slučajna promjena određenog dijela gena ima značajnu ulogu u radu svakog evolucijskog algoritma (uz iznimku evolucijskog programiranja koje se temelji samo na rekombinaciji) jer osigurava povezanost prostora pretrage kako je ranije rečeno. Međutim pitanje s početka ovog odjeljka, nema zadovoljavajućeg odgovora. Potrebno je prvo pogledati limese vjerojatnosti mutacije; kada vjerojatnost mutacije teži u vrijednost 0, tada mutacije nema, što u radu evolucijskog algoritma ima za posljedicu vrlo teško izvedivu konvergenciju algoritma. Druga je granica kada je vjerojatnost mutacije jednaka 1. U tom slučaju situacija je slična kao i kada je mutacija jednaka 0, odnosno, kada je mutacija jednaka 1 svaki lokus gena promijeniti će vrijednost iz 0 u 1, za slučaj binarnog kodiranja te će rezultat biti inverzija početnog gena. Stoga se općenito vrijednost mutacije mora nalaziti u intervalu $p_m = \langle 0,1 \rangle$. U velikoj većini slučajeva, konkretna vrijednost određuje se tako da se provede niz simulacija uz različite vrijednosti ovog parametra, te se odabere ona vrijednost uz koju se na nizu simulacija postižu najbolji rezultati evolucijskog algoritma. Međutim, noviji radovi [75] pokazuju kako statička vrijednost parametra mutacije postiže inferiorne rezultate u usporedbi s vjerojatnošću mutacije koja se tijekom procesa evolucije mijenja. Pokazano je da u početnoj fazi, kada je potrebno značajno pretraživati prostor rješenja, poželjno je implementirati izraženiju mutaciju, dok se u završnoj fazi rada algoritma, kada su rješenja evoluirana do područja bliskog optimalnom, vjerojatnost mutacije treba smanjiti. Ovo svojstvo ilustrirano je nizom simulacijskih eksperimenata prikazanih u ovoj disertaciji, pri čemu algoritam vrlo brzo napravi značajan napredak na skali dobrote, dok je za kasnije fino ugađanje rješenja za svega

nekoliko postotaka na skali dobrote, potrebno uložiti približno sličan broj generacija kao u prvoj fazi, gdje je promjena dobrote značajna. Ovo se objašnjava time da u toj završnoj fazi, stohastička priroda algoritma općenito, a posebice parametra mutacije zapravo ima štetne posljedice uništavajući dobra rješenja, kojima je potrebna vrlo mala prilagodba za dosizanje globalnog optimuma. Može se reći da je mutacija u toj završnoj fazi previše agresivna. U ovoj se disertaciji ispituje nekoliko pristupa poznatih u literaturi za ugađanje (off-line) i upravljanje (on-line) parametrom mutacije te se diskutiraju rezultati svakog pristupa. Kod ugađanja parametra mutacije, njegova je vjerojatnost neovisna o trenutnom stanju populacije i utjecaju tog parametra na vođenje evolucijskog procesa. Kod upravljanja parametrom mutacije, taj se parametar mijenja u ovisnosti o uspjehu populacije i njezinom odgovoru na određenu vrijednost mutacije. Najjednostavnija metoda ugađanja je, kao što je već rečeno, provođenje niza prethodnih eksperimenata i izbor određene vrijednosti koja se ne mijenja. Jedna metoda upravljanja je mijenjanje on-line parametra mutacije u ovisnosti o najboljem rješenju prisutnom u populaciji i standardnoj devijaciji populacije. Ukoliko standardan devijacija padne ispod određene razine, a globalni optimum nije pronađen, mijenja se vrijednost parametra mutacije s nadom u pronalazak novih rješenja, odnosno uvođenja novog genetskog materijala u populaciju [73]. Također zanimljivo novopredloženo rješenje je kodiranje vjerojatnosti mutacije u sami kromosom. Na taj način vjerojatnost mutacije postaje predmetom evolucije, oni kromosomi koji postižu bolje vrijednosti dobrote na implicitan način šire pripadajuću vjerojatnost mutacije u nadolazeće populacije.

Vjerojatnost križanja i veličina populacije su dva parametra za koja u literaturi ima vrlo malo dostupnih analiza koje se bave utjecajem ovih parametara na konvergenciju algoritma. Eksperimentalno su utvrđene vrijednosti ovih parametara u granicama $p_c = [0,4 - 0.8]$ i veličina populacije 30-100 jedinki, no ove rezultate treba uzeti s oprezom jer se odnose na jednu određenu skupinu problema, i njihova generalizacija nije dokazana. Zapravo, pokazano je da ne postoje robusni parametri evolucijskog algoritma, već su oni uvijek ovisni o specifičnostima problema na koji se EA primjenjuje. Tako će jedan algoritam s jednim parametrima biti uspješan na jednom određenom problemu, dok na sljedećem problemu njegov uspjeh ne mora biti adekvatno dobar, što je i dokazano *no free lunch* teoremom.

Iz prethodnih razmatranja slijedi da potraga za optimalnim parametrima općenitog evolucijskog algoritma završava u slijepoj ulici. Utemeljeno je uložiti određeni trud za ugađanje parametara algoritma za rješavanje specifičnog problema. Pri tome su parametri koji će se u ovoj disertaciji razmatrati sljedeći:

Tablica 3-2. Izabiranje parametara koevolucijskog algoritma.

Parametar	Metoda	Povratna veza
Mutacija	upravljanje / ugađanje	da / ne
Križanje	ugađanje	ne
Populacija	ugađanje	ne
Selekcija	ugađanje	ne

Tablicom [Tablica 3-2] prikazani su parametri i način njihovog ugađanja za koevolucijski algoritam.

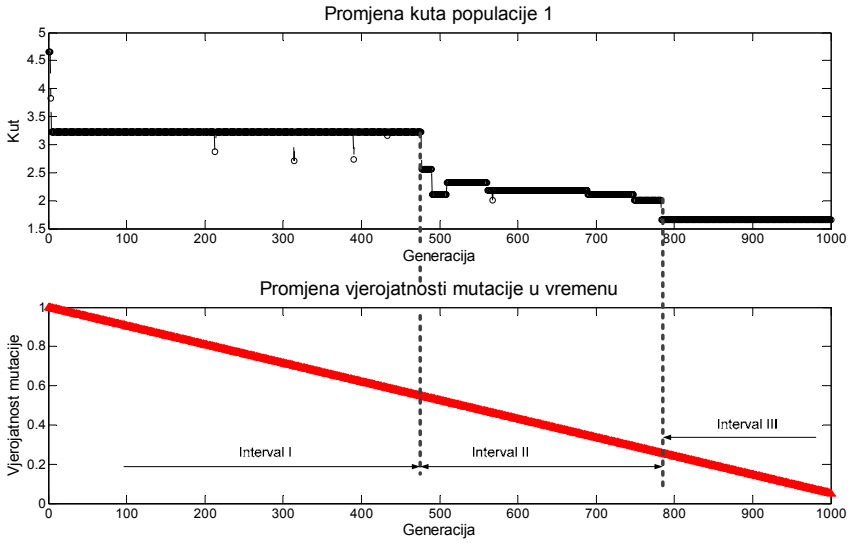
Kod prve metode, metoda M I, parametri su sljedeći: vjerojatnost križanja p_c = 0,7, vjerojatnost mutacije p_m = 0,1, veličina populacije M = 30, metoda selekcije je ruletno pravilo, bez skaliranja. Sve su simulacijske analize provedene s istim početnim i konačnim uvjetima dva koevoluirajuća agenta, prikazanima slikom [Slika 3-56]. Izboru prethodno provedenih parametara prethodila je serija simulacija, tako da su ovi parametri ugođeni i njihova se vrijednost tijekom simulacija nije mijenjala.

Kod druge metode, metoda M II, svi su parametri jednaki prethodnima, uz varijabilan parametar mutacije. Mutacija se kod metode M II mijenja prema sljedećem izrazu:

$$p_m = 1 - 0.9 \cdot \frac{t}{G}, \tag{74}$$

pri čemu je t oznaka trenutne generacije i t se mijenja od 1 do G, pri čemu je G maksimalno dopušteni broj generacija. Ova zakonitost predstavlja monotono opadajuću linearnu ovisnost vjerojatnosti mutacije o trenutnoj generaciji. Ova je implementacija motivirana idejom da na kraju evolucijskog procesa mutaciju treba smanjiti, kako je objašnjeno u razmatranju ranije.

U trećoj metodi, M III, diskretizira se ukupni broj iteracija te se pokušava, na temelju rezultata metode M II, odabrati takve diskretne vrijednosti parametra mutacije koje osiguravaju ukupnost konvergencije. Uspoređujući dijagram konvergencije i vremensku promjenu vrijednosti mutacije p_m pretpostavlja se da je moguće identificirati pri kojim vrijednostima mutacije se događaju povoljne promjene u konvergenciji, kako je prikazano slikom [Slika 3-64]. Motivacija je ove metode da se u algoritam implementira jedna ili više povoljnih stopa mutacije, za koje se identificira da u određenom trenutku rezultiraju povoljnom promjenom konvergencije.



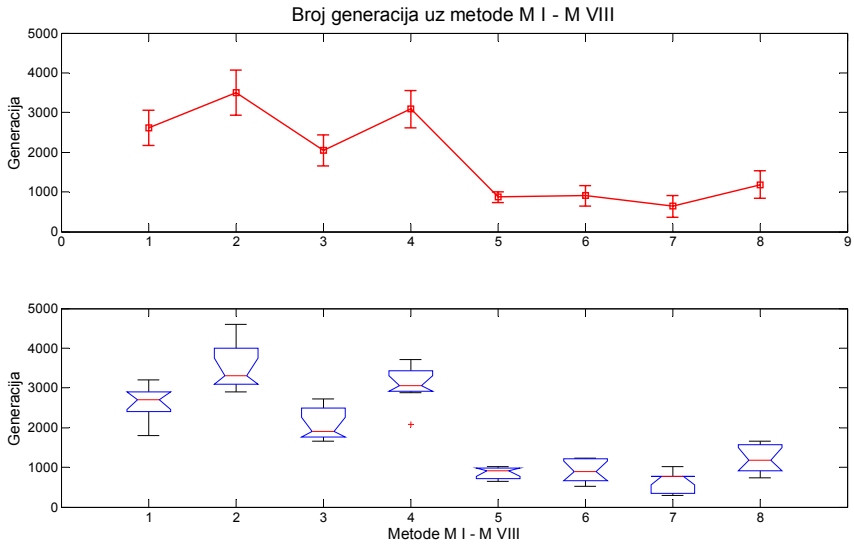
Slika 3-64. Identifikacija povoljne vjerojatnosti mutacije.

Na slici [Slika 3-64] prikazana je, radi jasnoće, samo najnepovoljnija komponenta konvergencije, u ovom slučaju to je kriterij minimizacije kuta. Ostale komponente konvergiraju ranije. Na gornjem su prikazu vidljiva tri izražena područja, redom intervali I, II i III. Vjerojatnost mutacije mijenja se prema izrazu definiranom u (74), što je prikazano nizom crvenih trokuta. Jasno je da obzirom na stohastičku prirodu algoritma, granice intervala I, II i III su varijabilne, no određene karakteristike su identificirane i metoda M III implementirana je kako slijedi: ukupan broj iteracija podijeljen je na tri. Diskretna vrijednost mutacije u svakom od intervala definirana je kao:

$$p_m^i = \frac{p_{\max}^i - p_{\min}^i}{2}, \tag{75}$$

odabirući tako srednju vrijednost mutacije za svaki interval. Pri čemu je p_m^i vjerojatnost mutacije u intervalu *i*, $p_{\max}^i, p_{\min}^i$ su najveća i najmanja vrijednost mutacije na granicama intervala *i*. I kod ove metode, ne mijenjaju se vrijednosti ostalih parametara evolucijskog algoritma.

Sljedeća metoda, M IV, je metoda temeljena na upravljanju vrijednošću mutacije na način da kada unutar 50 uzastopnih generacija algoritma nema promjene dobrote najboljeg člana populacije, mutacija se nasumično mijenja. Ukoliko unutar dodatnih pedeset iteracija nema poboljšanja, mutacija se mijenja, ali na takav način, da ukoliko se u prethodnom koraku povećala, sada se smanjuje za dvostruko veći iznos i obratno. Potrebno je uvijek voditi računa o tome da mutacija mora ostati u granicama između vrijednosti 0 i 1.



Slika 3-65. Konvergencija i razina pouzdanosti za različite metode definiranja parametara koevolucijskog algoritma.

Metode V – VIII istovjetne su metodama I – IV, uz tu razliku što je sada umjesto metode selekcije temeljene na ruletnom pravilu, izabrana metoda Pareto dominacije. Kod ove se metode, na temelju Pareto selekcije odabire skup najboljih rješenja koja nisu dominirana, vidjeti sliku [Slika 3-62], te se nedominirana rješenja bezuvjetno kopiraju u nadolazeću generaciju. Ovo je slično metodi elitizma kod jednostavnog genetskog algoritma i trebalo bi imati za posljedicu ubrzavanje konvergencije algoritma.

Slikom [Slika 3-65] prikazana je ovisnost broja generacija potrebnih za pronalaženje zadovoljavajućeg rješenja uz metode promjene vjerojatnosti mutacije ostalih parametara evolucijskog algoritma M I – M VIII. Proizlazi da je najveća dobit ostvarena nakon uvođenja selekcije temeljene na metodi Pareto dominacije, odnosno, prilikom čuvanja nedominiranih pojedinaca u populaciji. Također, „ručno“ ugađanje vjerojatnosti mutacije postiže usporedive rezultate kao i sofisticirane metode ugađanja temeljene na distribuciji populacije. Obzirom da vrijeme potrebno za takvo podešavanje parametra populacije nije predugo, potrebno je algoritam ispitati sa skupom diskretnih vrijednosti mutacije 0,01 – 0,2 u promatranom slučaju, upitno je isplati li se ulagati trud za implementaciju sofisticiranijih metoda obzirom na dobitak na strani ukupnog broja iteracija potrebnih za konvergenciju algoritma. Na slici [Slika 3-65] gornji prikaz pokazuje raspodjelu srednje vrijednosti za metode M I do M VIII, uz standardnu devijaciju (vertikalne linije) za svaku od metoda. Donji dijagram je tzv. box plot dijagram na kojem je prikazan donji kvartil, gornji kvartil, te medijan, uz raspodjelu ostalih vrijednosti

prikazanu vertikalnim linijama, tzv. ticalima, ili ako je razlika značajna, crvenim križićem. Vrijednosti su prikazane uz 20 simulacijskih eksperimenata uz svaku metodu, s time da su dvije najbolje i dvije najlošije vrijednosti odbačene za svaku od metoda.

4 Validacija i verifikacija predložene metodologije

4.1 Dvoručni robotski sustav Laboratorija za projektiranje izradbenih i montažnih sustava

Kako bi se predložena metodologija mogla konačno potvrditi, provedeni su inicijalni eksperimenti na opremi dostupnoj u Laboratoriju za projektiranje izradbenih i montažnih sustava Fakulteta strojarstva i brodogradnje, Sveučilišta u Zagrebu.



Slika 4-1. Roboti korišteni pri provedbi inicijalnih eksperimenata.

Odabrana su dva robota s po šest stupnjeva slobode gibanja proizvođača Fanuc, model Lr Mate 200iC.

Obzirom da su ovo roboti s više stupnjeva slobode gibanja nego što se razmatra u razvijenom modelu, svi zglobovi osim zglobova J_2 i J_3 držani su fiksnima, tako da se implementacija odvija u vertikalnoj ravnini.

Roboti su opremljeni senzorima momenta i sile, kamerama i odgovarajućom programskom podrškom primjenom koje je moguće ispitivati različite scenarije. Svaki robot je upravljan zasebnim računalom, što znači da ne postoji zajedničko procesorsko vrijeme i ne postoji a priori sinkronizacija između robota. Kako bi se ova sinkronizacija ostvarila, korišten je Fanucov dodatak imena Robot Link, pri čemu je korištena *master-slave* konfiguracija između dva robota.

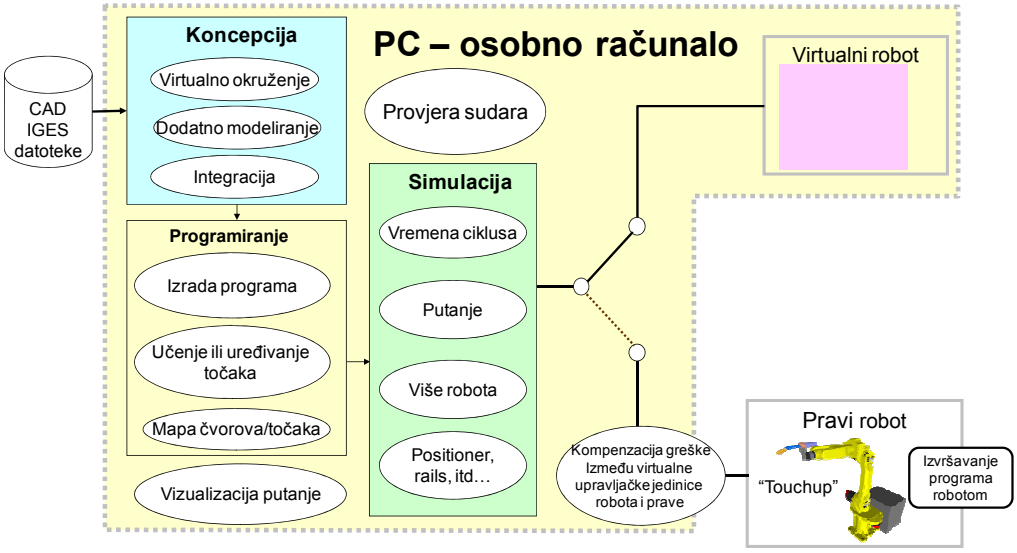
Upravljačko računalo robota koji je definiran kao *master* putem veze Fast Ethernetom permanentno šalje informacije o svojoj konfiguraciji robotu definiranom kao *slave*. *Slave* robot šalje potvrdni kod (*acknowledge, ACK*) *masteru*, nakon čega se gibanje nastavlja do sljedećeg komunikacijskog trenutka. Ukoliko se dogodi gubitak komunikacije između robota tijekom kretanja robota, uslijediti će trenutno zaustavljanje oba robota. Pored Ethernet komunikacije, koriste se i digitalne ulazno – izlazne linije radi usklađivanja kretanja. Njihovom primjenom moguće je prelaziti iz sinkroniziranog kretanja u nesinkronizirano i obratno.

Kod provedbe inicijalnog eksperimenta, točke dobivene evolucijskim algoritmom ručno su, putem privjeska za učenje, unesene u memorijske registre oba robota. Pri tome su početna i konačna točka putanje svake konfiguracije definirane kao *FINE* točke, odnosno referentne točke robota prolaze tim točkama. Točke koje se nalaze između rubnih točaka definirane su pomoću *CNT-XX* funkcije pri čemu *XX* označava razinu točnosti kojom će putanja biti interpolirana, odnosno roboti će u kontinuiranoj kretnji proći, ovisno o točnosti, blizu zadanih točaka ali će se pri tome uštediti vrijeme potrebno za zaustavljanje pri prolasku kroz točnu lokaciju točke.

4.2 Simulacija u fizičkom simulacijskom okružju Roboguide Fanuc Roboticsa

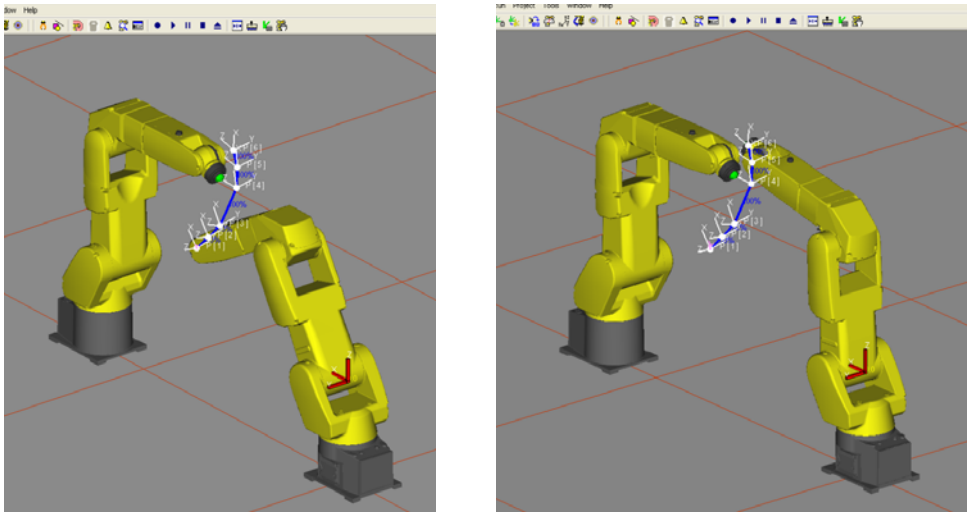
Simulacijsko okružje Roboguide fizičko je simulacijsko okružje koje pruža široke mogućnosti provedbe simulacija u realnim uvjetima. Osnovna namjena je brza prethodna verifikacija i validacija robotskog sustava uz minimalne utroške vremena i financijskih sredstava. Pri

modeliranju i kasnijoj verifikaciji u obzir su uzeta kašnjenja signala, komunikacije, izvedbe odgovarajućih kretanja. U naravi, razlika između programiranja na samom robotu ili u ovom sučelju minimalna je, osim što je jednostavnija komunikacija s ulazno izlaznim jedinicama robota u odnosu na stvarne robote gdje se sve odvija putem privjeska za učenje. Stoga samo programiranje robota za bilo koju kompleksniju namjenu nije preporučljivo izvoditi putem privjeska za učenje već putem Roboguide sučelja, te naknadno pohraniti program na upravljačko računalo samoga robota.U Roboguide sučelju moguće je vrlo detaljno definirati ne samo robota s kojim se radi nego i cijelu radnu okolinu u kojoj se robot nalazi. Pri tome su fizičke prepreke (poput ograda, objekata ili drugih robota) u ovome virtualnom svijetu prepoznate kao prepreke i predstavljaju područje nedostupno za robota. Postupak rada je sljedeći, u prvom se koraku odabire tip robota koji se ugrađuje u novu radnu okolinu. Definiraju se softverske karakteristike robota, i sva pripadajuća oprema i odgovarajući



Slika 4-2. Princip rada fanucovog *Roboguide* simulacijskog okružja.

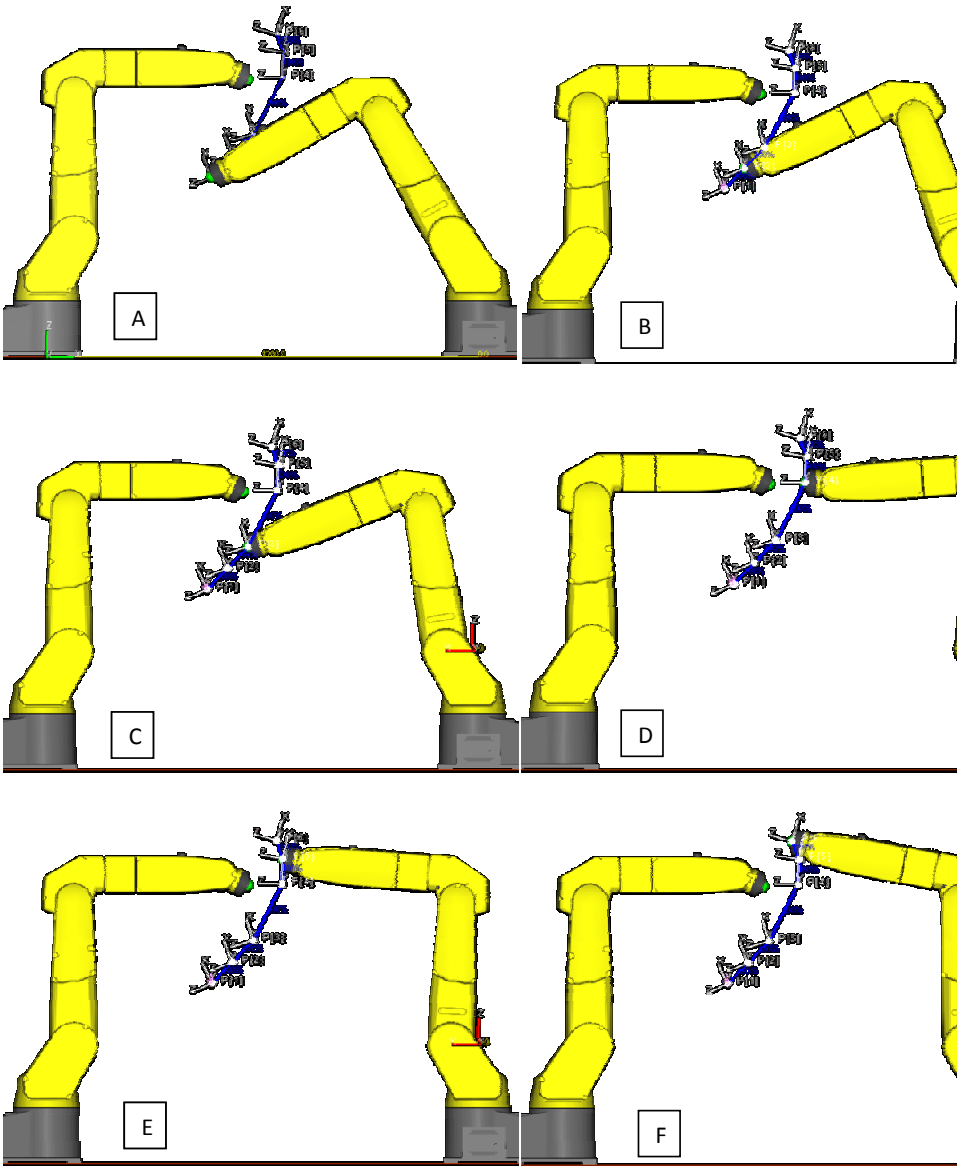
softverski moduli (kamera, senzor momenta i sile, ostali senzori). Nakon toga definira se vrsta komunikacije robota s perifernom opremom. Robot se potom dodaje u generičku okolinu gdje mu je potrebno definirati mjesto i orijentaciju u prostoru. Ukoliko je potrebno u okolinu ugraditi dodatne robote postupak se ponavlja.



Slika 4-3. Početna (lijevo) i konačna (desno) konfiguracija robota

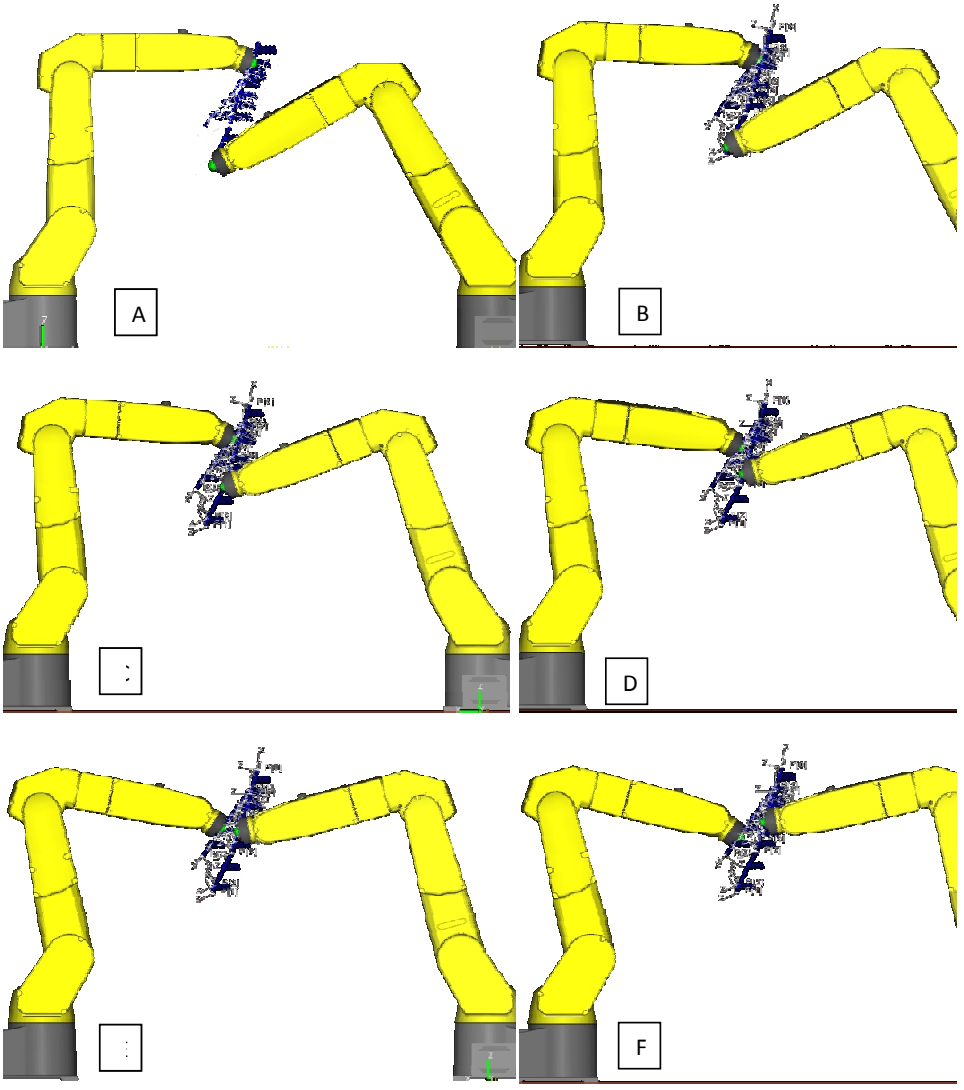
U slučaju obrađenom u ovoj disertaciji, obzirom na način na koji je postavljen model dvoručnog robota i odgovarajuće izlaze iz evolucijskog algoritma, koji su zakreti kutova u zglobovima, odabran je tzv *joint space* koordinatni sustav, odnosno koordinatni sustav zglobova kao onaj putem kojega će se definirati trenutni položaj robota. Iz ovog se koordinatnog sustava odgovarajućim transformacijama lako izračunava položaj bilo koje točke robota u nekom trenutku.Kao prvi slučaj, odabran je dvoručni robotski sustav pri čemu jedna ruka miruje. Lijeva robotska ruka u ovom slučaju miruje, predstavljajući statičku prepreku desnoj ruci, koja, obzirom na svoj početni i konačni položaj, treba izbjeći lijevu ruku. Ulazi u evolucijski algoritam su podaci o početnoj i konačnoj konfiguraciji svakog robota.

Izlaz iz evolucijskog algoritma je niz znakova, u ovome slučaju duljine 8, koji definira četiri među-konfiguracije (B-E) sa slike [Slika 4-4]. Kako je već u ranijem izlaganju objašnjeno, radi uštede procesorskog vremena i izbora odgovarajuće metode dekodiranja, početnu i konačnu konfiguraciju nije potrebno uvoditi u populaciju. Na slici je plavom bojom prikazana je trajektorija koju opisuje referentna točka na izvršnom članu robota. Primjećuje se da kriterij nejednolikosti profila brzine u gornjem slučaju nije dosegao globalni optimum.



Slika 4-4. Rješenje problema planiranja putanje za desnog robota. A-početna konfiguracija, F-ko konfiguracija.

Sljedeći korak predstavlja programiranje robota koje se izvodi na način da odgovarajućem koordinatnom sustaju zadaju referentne točke robota. Zadavajući niz t točaka, definira se kretanje robota iz jedne konfiguracije u drugu.



Slika 4-5. Rješenje problema planiranja putanje za oba robota. A-početna konfiguracija, F-ko konfiguracija.

Pri izvođenju eksperimenta pokazalo se da aproksimacija segmenata robota prav unosi određene nedoslijednosti u odnosu na realan model robota sastavljen od prost segmenata.

Robot je aproksimiran pravcima radi jednostavnosti provjere kolizija, iako bi moguća interpolacija npr. nizom kružnica odgovarajućeg promjera ili pravokutnikom, š

principom ne uzima se u obzir i poprečna dimenzija robota, iako se u značajnoj mjeri smanjuju nedosljednosti između simuliranog i realnog sustava. Druga je mogućnost modifikacija razmaka baznih segmenata dvaju robota koja u simulaciji treba biti manja u odnosu na stvarni razmak kod realnih robota.

Na razini implementacije potrebno bi bilo dodatno razviti protokole za komunikaciju između matlabovog ulazno – izlaznog sučelja i robotskih pozicijskih registara. Naime u ovoj se fazi izlazi evolucijskog algoritma ručno unose putem privjeska za učenje ili Roboguide sučelja u pozicijske registre robota.

Sljedeći slučaj za kojega će se rješenje prikazati u simulacijskom okrožju Roboguide odgovara scenariju *C* kod formalne analize dvoručnog robota. Ovo je općeniti slučaj kod kojega je potrebno prilagoditi kretanje dva robota u vremenu kako bi se izbjegla kolizija robota u bilo kojem trenutku vremena. Formalno je moguće da se roboti prilagođavaju jedan drugome na različite načine, no s aspekta utroška vremena i minimizacije ukupno prijeđenog puta svaki robot bi trebao odstupiti u određenom iznosu od najkraće moguće putanje. Primarni kriterij je kriterij izbjegavanja prepreka.

Tablica 4-1. Parametri početne i konačne konfiguracije lijevog robota.

Lijevi robot	J_1	J_2	J_3	J_4	J_5	J_6
Početna konfiguracija	0	0	0	0	-20	-145
Konačna konfiguracija	0	-17	-19	0	-20	-145

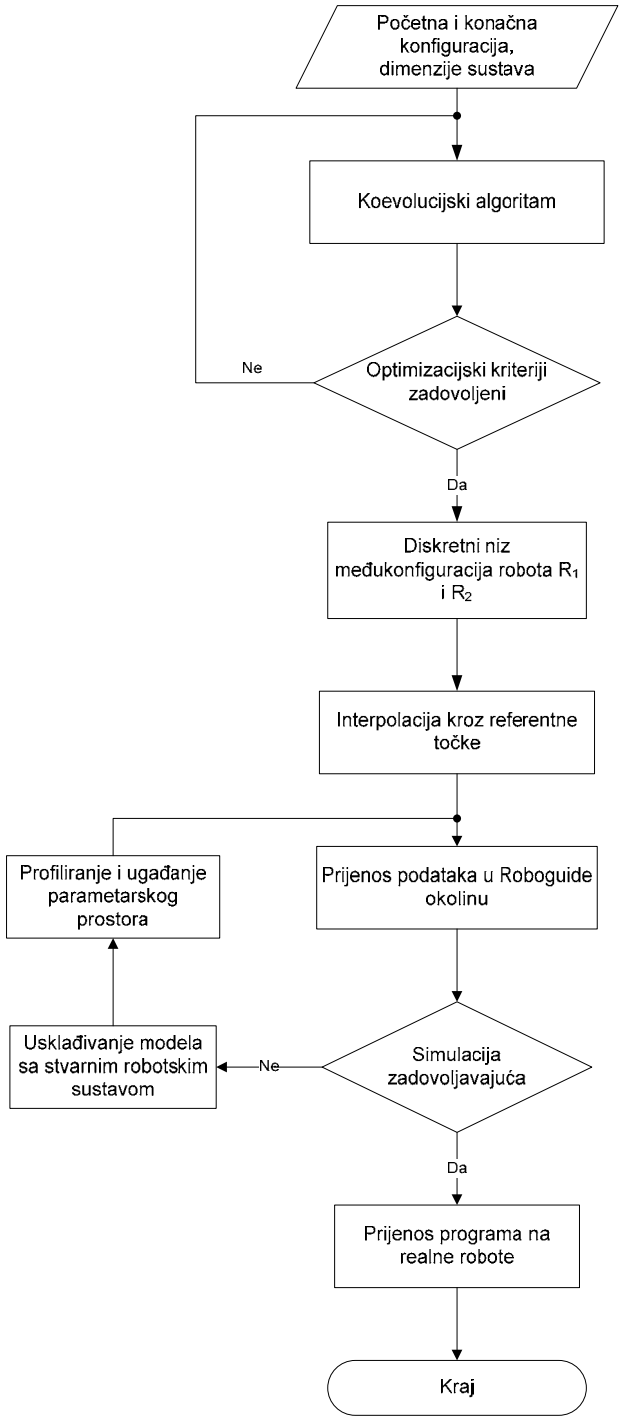
Tablicom [Tablica 4-1] prikazana je početna i konačna konfiguracija lijeve ruke dvoručnog robotskog sustava. Smisao oznaka je sljedeći J_1 – J_6 su standardno oznake odgovarajućeg zgloba robota počevši s baznim zglobom i završivši sa zglobom izvršnog člana na kojem se nalazi alat. Vrijednosti zakreta određenog kuta izražene su u stupnjevima u prostoru zakreta zglobova. Potrebno je primijetiti da se upravljanje izvodi samo sa dva zgloba, J_2 i J_3, što odgovara modelu razmatranom u ovoj disertaciji. Tablicom [Tablica 4-2] prikazani su odgovarajući parametri konfiguracije desne ruke dvoručnog robotskog sustava.

Tablica 4-2. Parametri početne i konačne konfiguracije desnog robota

Desni robot	J_1	J_2	J_3	J_4	J_5	J_6
Početna konfiguracija	0	30	-30	3	5	7
Konačna konfiguracija	0	9	3,2	3	5	7

I u ovom se slučaju problem koordinacije kretanja dvaju robota izvodi upravljanjem zglobova J_2 i J_3.

Podatci iz gornjih tablica predstavljaju ulaze u evolucijski algoritam, dok su izlazi iz evolucijskog algoritma međustanja konfiguracija robota lijeve i desne ruke. Slikom [Slika 4-5] prikazano je rješenje dobiveno evolucijskim algoritmom implementirano na dvoručni robotski sustav u sustavu Roboguide. Putanja je diskretizirana s četiri međukonfiguracije, što uz početnu i konačnu čini ukupno šest konfiguracija robota.



Slika 4-6. Validacija predložene metodologije.

Sljedeća, i ujedno konačna faza u provedbi verifikacije i validacije metodologije može biti implementacija programa napisanog u sustavu Roboguide na realnom robotskom sustavu. kako je već rečeno roboguide je realističan fizički simulator potpuno kompatibilan s realnim robotima. Programi napisani u njemu, nakon prijenosa odmah su primjenjivi na stvarnom robotskom sustavu.

Slikom [Slika 4-6] prikazane su faze provedbe verifikacije predložene metodologije. Početak predstavljaju željena stanja robota u početnom i konačnom trenutku. Nakon dobivanja zadovoljavajućih rješenja iz evolucijskog algoritma, primjenjuje se po potrebi razvijena metoda za rekonstrukciju kontinuirane putanje iz dobivenog diskretnog niza konfiguracija. Sljedeći korak je prijenos podataka o konfiguracijama u obliku zakreta zglobova u Roboguide programsko sučelje. Ovo se može izvoditi automatski putem TCP IP protokola, dok je u ovoj fazi istraživanja taj korak izvođen ručno.

Nakon simulacije u Roboguide okolini otklanjaju se nedostatci u vidu dimenzije nekonzistentnosti uslijed pojednostavljenog modela iz prvog koraka. najčešće je dovoljno povećati udaljenost između dva robota. Konačno, ovisno o raspoloživoj opremi moguće je prenijeti cjelokupni program iz simuliranog upravljačkog računala na stvarno računalo koje upravlja robotom i provesti eksperiment i na realnoj opremi. Potrebno je napomenuti da kod upravljanja realnim robotima bilo putem Roboguide sučelja, bilo direktno, prilikom kretanja robota potrebno je definirati niz kriterija. Ovi se kriteriji tiču koordinatnog sustava, načina kretanja robota te brzina i ubrzanja referentne točke robota. Potpuna sigurnost pri izbjegavanju kolizija osigurana je ako se kroz sve referentne konfiguracije dobivene evolucijskim algoritmom robot kreće na način da se svaka od tih konfiguracija stvarno ponovi, odnosno da robot određeni vremenski period zauzme konfiguraciju pronađenu evolucijskim algoritmom. Na razini implementacije ovo se izvodi tako da se sve konfiguracije definiraju naredbom *FINE* pri programiranju, no pokazalo se da u tom slučaju može doći do nejednolikog, isprekidanog kretanja robota. Stoga je moguće međukonfiguracije programirati s određnim, malim odstupanjem naredbom *CNT*, kako bi kretanje bilo jednoliko tijekom vremena, no nije moguće unaprijed definirati koliko je dopušteno odstupanje u odnosu na putanju dobivenu evolucijskim algoritmom.

5 Zaključak

U ovoj disertaciji razvijena je metodologija planiranja kretanja za dva robota $2R$ konfiguracije koji dijele radni prostor temeljena na koevolucijskom algoritmu. U prvom dijelu disertacije prikazane su metode planiranja kretanja za jednog agenta u radnom prostoru te su istaknute prednosti i nedostaci metoda dostupnih u literaturi. Potom su prikazane metode planiranja u višeagentskom okružju gdje su istaknuti problemi planiranja obzirom na kombinatornu složenost prirode problema planiranja.

Obzirom da se pristup planiranju putanje u ovoj disertaciji temelji na heurističkoj metodi evolucijskih algoritama, ukratko su prikazane četiri osnovne (kanonske) klase evolucijskih algoritama. Pokazane su dosadašnje spoznaje o računskoj složenosti evolucijskih algoritama, odnosno razmatrano je vrijeme potrebno za konvergenciju evolucijskog algoritma. Model dvoručnog robota implementiran je sekvencijalno. Optimizacijski kriteriji definirani su i implementirani postupno, kako se pojavila potreba za njima. Pokazalo se da se sa skupom od četiri kriterija mogu razviti zadovoljavajuća ponašanja agenata, uz prihvatljivo dugo vrijeme potrebno za konvergenciju. Iako se prikazana metodologija pokazala neprikladnom za upotrebu u on-line načinu rada stvarnog robotskog sustava, jedna od prednosti je ta, da ukoliko se rješenje ne može pronaći, to se pokazuje u relativno ranoj fazi evolucije.

Nakon definicije skupa kriterija, model je proširen na sustav od dva agenta, pri čemu agenti dijele radni prostor i predstavljaju dinamičku prepreku jedan drugome. Obzirom da ovako definirani problem postaje problem višekriterijskog optimiranja s nejasnim vezama između pojedinih optimizacijskih kriterija, analizirane su veze među pojedinim kriterijima. Također je predložena metoda izbora temeljena na Pareto dominaciji koja se pokazala značajno boljom od klasičnog izbora temeljenog na samoj dobroti pojedinaca.

Sustavno je analiziran odnos parametara evolucijskog algoritma, poput veličine populacije, vjerojatnosti mutacije, vjerojatnosti križanja, metode izbora te su određene optimalne kombinacije ovih parametara za prikazani problem.

Metodologija je verificirana na većem nizu simulacijskih eksperimenata, pri čemu se ustanovilo da parametri evolucijskog algoritma nisu neovisni o početnim i konačnim uvjetima dva agenta, iako se vrednovanje izvodi istom funkcijom dobrote. Dvoručna robotska konfiguracija je formalno analizirana te su izvedeni analitički izrazi za provjeru kolizija direktno putem kutova zakreta u zglobovima. U odnosu na pristup gdje se kolizije provjeravaju traženjem sjecišta odgovarajućih pravaca, vremenska ušteda nije zanemariva. Provjera putem sjecišta pravaca ipak ostaje potrebna prilikom provjere kolizije u međustanjima, kada se agent pomiče iz prethodne u sljedeću konfiguraciju. Također je dokazano, da uz određena logična geometrijska ograničenja, uvijek je moguće pronaći takvu međukonfiguraciju koja rezultira neprekinutim gibanjem od prethodne do sljedeće konfiguracije, pri čemu je putanja referentne točke na izvršnom članu robota opisana po dijelovima linearnom funkcijom klase C_0.

Obzirom na vremensku složenost promatranog problema planiranja kretanja predložena je metoda diskretizacije radnog prostora, te su izračunate najveće dopuštene vrijednosti kutova zakreta u zglobovima u ovisnosti o kinematskim ograničenjima odgovarajućeg realnog robotskog sustava. U ovisnosti o vrijednostima ovih kutova, izračunata je minimalna duljina gena za najnepovoljniji slučaj. Predložena je metoda kojom je moguće iz diskretnog niza konfiguracija, putem interpolacije vremenske ovisnosti zakreta kutova u zglobovima, pronaći niz konfiguracija s po volji malim vremenskim intervalom. Predloženo je i validirano osam metoda za ugađanje i upravljanje parametrima evolucijskog algoritma te su istaknute one metode koje na promatranom problemu postižu najbolje rezultate.

Razvijena metodologija verificirana je, s ciljem potvrde hipoteze u fizičkom simulacijskom okružju Roboguide u kojemu je moguće vjerno, uz sva ograničenja iz realnog svijeta provoditi različite robotske zadaće. Utvđena su ograničenja primjene predložene metodologije u prvom redu zbog aproksimacije robota pravcima, što ne odgovara stvarnom stanju gdje segmenti stvarnih robota imaju trodimenzionalni karakter. Ovu činjenicu bilo bi potrebno ugraditi u model dvoručnog robota razvijenom u poglavlju 3.2. Aproksimaciju pravcem moguće je zamijeniti pravokutnikom za razmatranje dvodimenzionalnih problema te kvadrom ili cilindrom za trodimenzionalni slučaj. Ovo proširenje ne bi imalo utjecaj na ukupnu konvergenciju evolucijskog algoritma, već samo na vrijeme potrebno za pronalaženje

zadovoljavajućeg optimuma odnosno putanje. Prilagodba je sada vršena na način da su bazni segmenti robota u fazi modeliranja postavljeni s manjim razmakom u odnosu na bazne segmente realnih robota čime su se postigli zadovoljavajući rezultati.

Obzirom da se planiranje putanje izvodi u Matlabovom okružju, dok se programiranje robota izvodi u specifučnom jeziku Karel, izmjena podataka o konfiguracijama izvodi se jednokratno i to nakon što je planiranje u prvoj fazi uspješno završeno. Komunikaciju je moguće izvoditi ručno, uz upisivanje zakreta svakog zgloba u odgovarajuće memorijske registre robota, što je u ovoj fazi učinjeno, dok se predviđa i razvoj komunikacijskog protokola za izmjenu podataka putem standardnog TCP/IP protokola, koji je podržan na obje platforme.

Motivacija za ovo istraživanje bila je, između ostalog, i mogućnost implementacije robotske konfiguracije koja se nalazi u Laboratoriju za projektiranje izradbenih i montažnih sustava na rješavanje problema tipa *Bin picking*. Ovaj naziv odnosi se na probleme kada je robotom potrebno iz određenog prostora, koji sadrži objekte razmještene po slučaju, uzimati hvataljkom robota željeni objekt, bez kolizije s preprekama iz okoline ili s drugim objektima. Ovaj problem proučava se već relativno dugo i danas se uspješno rješava 3D vizijskim sustavom i jednom robotskom rukom [76], [77]. Međutim, u slučaju kada dva robota imaju pristup području u kojem se nalaze objekti, potrebno je odgovoriti, između ostalog, na probleme planiranja akcija, prioriteta robota, ali, isto tako i dati rješenje za planiranje kretanja koje ne rezultira kolizijama pri istodobnom uzimanju objekata sa dva robota.

Daljna istraživanja moguća su u sljedećim pravcima:

- na razini evolucijskih algoritama moguće je daljnje ubrzavanje rada algoritma temeljeno na heurističkom inicijaliziranju populacije. Dobri pojedinci, odnosno rješenja u obliku trajektorija mogu se čuvati i, u slučaju da nepoznati problem ima sličnosti s prethodno riješenim problemom, ugraditi direktno u početnu populaciju rješenja. Motivacija je ubrzavanje pronalaska zadovoljavajućeg optimuma.
- Na razini modela, moguće je proširenje na puni trodimenzionalni prostor. Određena ograničenja u razvijenoj metodologiji proizlaze iz pojednostavljenog modela dvoručnog robota modeliranog pravcima. Vjernija slika dobila bi se kada bi segmenti bili pravokutnici, ili nizovi kružnica za ravninski, te kvadrovi ili cilindri za trodimenzionalni prostor.

- Kod implementacije metodologije na realni robotski sustav, s ciljem potpune automatizacije procesa planiranja potrebno je razviti odgovarajuće programsko rješenje za komunikaciju između osobnog računala i upravljačkog računala robota, pri čemu je sama komunikacija izvediva putem standardnog mrežnog TCP/IP protokola.
- Pokazalo se da je ukupnost konvergencije algoritma obrnuto proporcionalna broju optimizacijskih kriterija. Prostor pretrage, kao i njegova složenost raste uvođenjem dodatnih kriterija. Obzirom da je odabran najmanji skup od četiri kriterija kojima je moguće razviti odgovarajuće kretanje dvaju robota, a da vrijeme potrebno za pronalaženje rješenja značajno raste u odnosu na skup od 3 kriterija (kada nema prisutne prepreke u prostoru), te da se ukupnost konvergencije smanjuje, moguće je traženje povećanosti ukupnosti konvergencije hibridizacijom predložene evolucijske metode s drugim optimizacijskim metodama lokalnog karaktera, temeljenih na gradijentu prostora pretrage.
- Standardno se kod optimizacije evolucijskim algoritama javlja problem izbora odgovarajućih parametara algoritma. U ovom radu predloženo je i evaluirano osam različitih metoda izbora ovih parametara. Ovo je područje istraživanja evolucijskih algoritama danas vrlo aktivno te je moguće daljnje unapređivanje predloženih metoda kao i implementacija metoda dostupnih u recentnoj literaturi [78], [79].

Zaključno, može se reći da je hipoteza o mogućnosti planiranja putanje dvoručnog robota primjenom razvijenog evolucijskog algoritma potvrđena. Ipak, obzirom na stohastičku prirodu evolucijskih algoritama, ne postoji formalni dokaz o konvergenciji algoritma u globalno optimalno područje. Povoljna okolnost je to da su često prihvatljiva i suboptimalna rješenja algoritma, uz ograničenje na kriterij kolizija, koji uvijek mora biti u potpunosti zadovoljen.

Na temelju izloženog, proizlaze sljedeći znanstveni doprinosi:

- postavljanje modela, analiza i razvoj simulacijskog okružja dvoručnog robotskog sustava.
- razvoj koevolucijskog optimizacijskog algoritma za upravljanje dvoručnim robotskim sustavom
- utvrđivanje prirode veza među optimizacijskim kriterijima i parametrima evolucijskog algoritma

- prijedlog i verifikacija modificiranih metoda ugađanja parametara evolucijskog algoritma
- razvoj generičkog tehničkog sustava sastavljenog od razvijenog koevolucijskog algoritma i dva robota 2*R* konfiguracije.

6 Popis literature

[1] E. Guizo, "Building Andorids to Understand Humans - Hiroshi Ishiguro," *IEEE Spectrum*, vol. 47, no. 4, pp. 41-44, May 2010.

[2] K. A. De Jong, "Evolving Intelligent Agents: A 50 Year Quest," *IEEE Computational Intelligence Magazine*, vol. 3, no. 1, pp. 12-17, February 2008.

[3] J. Canny, *The Complexity of Robot Motion Planning*. Cambridge: MIT Press, 1988.

[4] J. Canny, "On Computability of Fine Motion Plans," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, 1989, pp. 177-182.

[5] J. van Leeuwen, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*.: MIT Press, 1990.

[6] J.-C. Latombe, *Robot Motion Planning*. Boston, USA: Kluwer Academic Publishers, 1991.

[7] Y. K. Hwang and N. Ahuja, "Gross Motion Planning - A Survey," *ACM Computing Surveys*, no. 3, pp. 219-291, 1992.

[8] J.-C. Latombe, "Motion Planning: A Journey of Robots, Molecules, Digital Actors and Other Artifacts," *International Journal of Robotics Research*, vol. 18, no. 11, pp. 1119-1128, 1999.

[9] M. Sharir, "Algoritmic Motion Planning," in *Handbook of Discrete and Computational Geometry*, O'Rourke J. and J. E. Goodman, Eds. Boca Raton, Florida: CRC Press, 1997, pp. 733-754.

[10] P. Isto. (2003) Adaptive probabilistic roadmap construction with multi-heuristic local planning. Ph.D. Thesis.

[11] D. E. Koditscheck and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics*, vol. 11, pp. 412-442, 1990.

[12] R. A. Brooks and M. A. Flynn, "Fast, Cheap and Out of Control: A Robot Invasion of the Solar System," *Journal of the British Interplanetary Society*, vol. 42, pp. 478-485, 1989.

[13] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, Massachusetts: The MIT Press, 1998.

[14] A. Ram, R. Arkin, G. Boone, and M. Pearce, "Using Genetic Algorithms to Learn Reactive Control Parameters for Autonomous Robotic Navigation," *Journal of Adaptive Behavior*, vol. II, no. 3, pp. 277-305, 1994.

[15] C. Hocaoglu and A. C. Sanderson, "Multi-dimensional path planning using evolutionary computation," in *The 1998 IEEE International Conference on Evolutionary Computation*, Anchorage, 1998, pp. 165-170.

[16] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Learning*, 25th ed.: Addison Wesley, 2003.

[17] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Inettligence*.: Morgan Kaufmann Publishers, 2001.

[18] J. Kasać and B. Novaković, "Optimal Robot Control with Unspecified Initial and Final Conditions," in *Proceedings of 9-th Mediterranean Conference on Control and Automation*, Dubrovnik, 2001.

[19] A. Saffiotti, "The uses offuzy logic in autonomous robot navigation," *Soft Computing*, vol. I, no. 4, pp. 180-197, 1997.

[20] J. Yu, Q. Zhang, V. Kroumov, S. Cheng, and Z. Zhang, "Path Planning Algorithm for Robot in 3D Environment Based on Neural Network," in *Intelligent Robotics and Applications*. Berlin / Heidelberg: Springer, 2008, pp. 1081-1088.

[21] P. Isto and M. Saha, "Multi-Robot Path Planning by Incrementtal Coordination," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, 2006.

[22] B. Aronov, M. de Berg, A. F. Van der Stappen, P. Svetska, and J. Vleugesl, "Motion Planning for Multiple Robots," *Discrete and Computational Geometry*, vol. 22, pp. 505-525, 1999.

[23] K. G. Kant and S. W. Zucker, "Towards efficient trajectory planning: path velocity decomposition," *International Journal of Robotics Research*, vol. 5, pp. 72-89, 1986.

[24] P. A. O'Donnell and T. Lozano-Perez, "Deadlock-free and collision-free coordination of two robot manipulators," in *Proceedings of the IEEE International Conference of*

Robotics and Automation, 1989, pp. 484-489.

[25] G. Sanchez and J.-C. Latombe, "Using a PRM planner to compare centralized and decoupled planning for multi-robot systems," in *Proceedings of the IEEE Conference on Robotics and Automation*, 2002, pp. 2112-2119.

[26] A. S. Rana and M. S. Zalzala, "An Evolutionary Algorithm for Collision Free Motion Planning of Multi-Arm Robots," in *IEEE Genetic Algorithms in Engineering Systems: Innovations and Applications*, 1995.

[27] A. S. Rana and M. S. Zalzala, "An Evolutionary Plannerfor Near Time-Optimal Collison Free Motion of Multi-Arm Robotic Manipulators," in *UKACC International Conference on Control*, 1996.

[28] P. G. Devendra and K. Manish, "Optimization techniques applied to multiple manipulators for path planning and torque minimization," *Engineering Applications of Artificial Intelligence*, vol. 15, pp. 241-252, 2002.

[29] M. Ivanescu and N. Bizdoaca, "A two level hierarchical fuzzy controller for hyperredundant cooperative robots.," in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, 2000, pp. 3170-3175.

[30] M. Venegas and M.-R. Raymundo, "An Evolutionary Path Planner for Multiple Robot Arms," in *Evo Workshops, LNCS*, Heidelberg, 2009.

[31] H. Choset, *Principles of Robot Motion: Theory, Algorithms and Implementations*.: MIT Press, 2005.

[32] J. H. Reif, "Complexity of the mover's problem and generalizations," in *20th IEEE Symposium on Foundations of Computer Science*, 1979, pp. 421-427.

[33] X. Chen and Y. Li, "Smooth Formation Navigation of Multiple Mobile Robots," *International Journal of Control, Automation, and Systems*, vol. IV, no. 4, pp. 466-479, August 2006.

[34] R. Sexton and J. Gupta, "Comparative evaluation of genetic algorithm and backpropagation for training neural networks," *Information Sciences*, no. 129, pp. 45-59, 2000.

[35] J. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, Ma: MIT Press, 1992 (1975).

[36] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer Verlag, 1992.

[37] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149-172, 2000.

[38] T. Bäck, *Evolution Strategies Evolutionary Programming Genetic Algorithms*. New York: Oxford University Press, 1996.

[39] A. J. Keane and S. M. Brown, "The design of a sattelite boom with enhanced vibration performance using genetic algorithm techniques," in *Proceedings of the Conference on Adaptive Computing in Engineering Design*, Plymouth, 1996, pp. 107-113.

[40] P. Smolensky, "On the proper treatment of connectionism," *Behavioral and Brain Sciences*, vol. 11, no. 2, pp. 1-14, 1988.

[41] P. Ćurković and B. Jerbić, "Genetic algorithm for robot path estimation," in *Proceedings of the DAAAM*, Wien, 2007.

[42] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed., G. Rozenberg, Ed. Heidelberg: Springer Verlag, 2007.

[43] P. W. Price, *Biological Evolution*.: Saunders College Publishing, 1998.

[44] D. Hillis, "Co-evolving parasites improve simulated evolution as an Optimization procedure," *Artificial life II, SFI Studies in the Sciences of Complexity*, vol. 10, pp. 313-324, 1991.

[45] K. O. Stanley and R. Miikulainen, "Coevolution through evolutionary complexification," University of Texas at Austin, Austin, Tchnical report AI-02-298, 2000.

[46] R. P. Wiegand. (2003) An Analysis of Cooperative Coevolutionary Algorithms. Ph.D. Thesis, George Mason University.

[47] S. G. Ficici. (2004) Solution concepts in coevolutionary algorithms. Ph.D. Thesis, Brandeis University.

[48] P. S. Oliveto, J. He, and X. Yao, "Time Complexity of Evolutionary Algorithms for Combinatorial Optimization: A Decade of Results," *International Journal of Automation and Computing*, vol. 4, no. 3, pp. 281-293, July 2007.

[49] T. Bäck, D.B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. Bristol: IOP Publishing Ltd., 1997.

[50] G. Rudolph, "Convergence Analysis of Canonical Genetic Algorithms," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 96-101, 1994.

[51] G. Rudolph, "Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon," *Fundamenta Informaticae*, vol. 5, no. 1, pp. 96-101, 1998.

[52] H. Aytug and G. J. Koehler, "Stopping Criteria for Finite Length Genetic Algorithm," *ORSA Journal on Computing*, vol. 8, no. 2, pp. 67-89, 1996.

[53] M. Safe, J.A. Carballido, I. Ponzoni, and N.B. Brignole, "On Stopping Criteria for Genetic Algorithms," in *17th Brazilian Symposium on Artificial Intelligence*, 2004, pp. 405-413.

[54] S. Droste, T. Jansen, and I. Wegener, "On the Analysis of the (1+1) Evolutionary Algorithm," *Theoretical Computer Science*, vol. 276, no. 1-2, pp. 51-81, 2002.

[55] I. Wegener, "On the Design and Analysis of Evolutionary Algorithms," in *Proceedings of the Workshop on Algorithm Engineering as a new Paradigm*, Kyoto, 2000, pp. 37-47.

[56] J. Pettersson. (2006) Generation and Organization of Behaviors for Autonomous Robots. Ph.D. Thesis, Chalmers University of Technology.

[57] D. Floreano and S. Nolfi, *Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-organizing Machines*.: MIT Press, 2000.

[58] G. P. Wagner and L. Altenberg, "Complex adaptations and the evolution of evolvability," *Evolution*, vol. 50, pp. 967-976.

[59] F. Mondada and D. Floreano, "Evolution of neural control structures: some experiments on mobile robots," *Robotics and autonomous systems*, vol. 16, no. 2-4, pp. 183-195, 1995.

[60] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179-211, 1990.

[61] G. S. Hornby, M. Fujita, S. Takamura, T. Yamamoto, and O. Hanagata, "Autonomous evolution of gaits with the sony quadruped robot," in *Proceedings of the Genetic and Evolutionary Computation conference (GECCO)*, 1999, pp. 1297-1304.

[62] P. Nordin and W. Banzhaf, "An on-line method to evolve behavior and to control a miniature robot in real time with genetic programming," *Adaptive Behavior*, vol. 5, no. 2, pp. 107-140, 1997.

[63] P. Augustsson, K. Wolff, and P. Nordin, "Creation of a learning, flying robot by means of evolution," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, 2002, pp. 1279-1285.

[64] S. Watson, S. Ficici, and J. Pollack, "Embodied evolution: Distributing an evolutionary algorithm in a population of robots," *Robotics and Autonomous Systems*, vol. 39, no. 1,

pp. 1-18, 2002.

[65] C. A. C. Coello, "A Comperhensive Survey of Evolutionary-Based Multiobjective Optimization Techniques," *Knowledge and Information Systems*, vol. 1, no. 3, pp. 269-308.

[66] D. A. Van Veldhuizen. (1999) Multiobjective Evolutionary Algoriths: Classifications, Analyses, and New Innovations. Ph.D. Thesis, Air Force Institute of Technology, Ohio.

[67] Y. Fei, D. Fuqiang, and Z. Xifang, "Collision-free motion planning of dual-arm reconfigurable robots," *Robotics and Computer-Integrated Manufacturing*, vol. 20, pp. 351-357, 2004.

[68] F. Yano and Y. Toyoda, "Preferable movement of multi-joint robot arm using genetic algorithm," in *SPIE Conference on Intelligent Robots and Computer Vision*, Boston, 1999, pp. 80-88.

[69] Y. Toyoda and F. Yano, "Optimizing Movement of A Multi-Joint Robot Arm with Existence of Obstacles Using Multi-Purpose Genetic Algorithm," *Industrial Engineeing & Management Systems*, vol. 3, no. 1, pp. 78-84, 2004.

[70] L. Tian and C. Collins, "An effective trajectory planning method using a genetic algorithm," *Mechatronics*, vol. 14, pp. 455-470, 2004.

[71] E. J. Pires Solteiro, J. A. Tenreiro Machado, and P. B. Moura Oliveira, "Robot Trajectory Planning Using Multi-objective Genetic Algorithm Optimization," in *Proceedings of the GECCO 2004*, 2004, pp. 615-626.

[72] M. Wooldridge, *An Introduction to MultiAgent Systems*. Chichester: John Wiley & Sons, 2006.

[73] P. Ćurković, B. Jerbić, and T. Stipančić, "Hybridization of adaptive genetic algorithm and ART 1 neural architecture for efficient path planning of a mobile robot," *Transactions of FAMENA*, vol. 32, no. 2, pp. 11-21, 2008.

[74] P. Ćurković, B. Jerbić, and T. Stipančić, "Swarm-based Approach to Path Planning Using Honey-bees Mating Algorithm and ART Neural Network," *Solid State Phenomena*, vol. 147-149, pp. 74-80, 2009.

[75] S. K. Smit and A. E. Eiben, "Parameter Tuning of Evolutionary Algorithms:," in *EvoStar 2010*, Istambul, 2010.

[76] K. Ohba and K. Ikeuchi, "Recognition of the multi specularity objects for bin-picking task ," in *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96,* Osaka, 1996, pp. 1440-1447.

[77] O. Jong-Kyu, B. Kyeong, K. Daesik, and L. Sukhan, "Development of structured light based bin picking system using primitive models," in *IEEE International Symposium on Assembly and Manufacturing, 2009. ISAM 2009.*, Suwon, 2009, pp. 46-52.

[78] S. K. Smit and A. E. Eiben, "Parameter Tuning of Evolutionary Algorithms: Generalist vs. Specialist," in *Lecture Notes in Computer Science*, 2010, pp. 542-551.

[79] S. K. Smit and A. E. Eiben, "Using Entropy for Parameter Analysis of Evolutionary Algorithms," *Empirical Methods for the Analysis of Optimization Algorithms*, pp. 211-224, 2009.

[80] (2010, May) IFR International Federation of Robotics. [Online]. http://www.ifr.org/

Životopis

Ime i prezime:	Petar Ćurković
E-mail:	petar.curkovic@fsb.hr
Datum rođenja:	02.12.1977.
Mjesto rođenja:	Zagreb, Hrvatska
Školovanje:	2005. - 2010., poslijediplomski studij strojarstva, Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu 1996. - 2004., dodiplomski studij strojarstva, Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu 1992. - 1996., Srednja tehnička škola, Zagreb 1984. - 1992., Osnovna škola, Zagreb
Zaposlenje:	01.01.2005. - danas, Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu, Zavod za robotiku i automatizaciju proizvodnih sustava, Katedra za projektiranje izradbenih i montažnih sustava, znanstveni novak - asistent 03. 04. 2004. - 31. 12. 2004., Yazaki Europe Ltd
Znanje jezika	Govori i piše engleski i njemači jezik
Sudjelovao na znanstveno-istraživačkim projektima:	2009. -, istraživač na projektu ″ Improving GRAsping Movements by predictions based on Observation″, UKF projekt podržan od Ministarstva znanosti, obrazovanja i športa RH, glavni istraživač prof. dr. sc. Danica Kragić. 2008. -, istraživač na tehnologijskom projektu ″ Automatska montaža dvoručnim robotom, TP-08/0120-53projekt podržan od Hrvatskog instituta za tehnologiju, glavni istraživač prof. dr. sc. Bojan Jerbić. 2007. -, istraživač na projektu ″Autonomna višeagentna automatska montaža″, znanstveni projekt br. 120-1201948-1941 podržan od Ministarstava znanosti obrazovanja i športa RH, glavni istraživač prof. dr. sc. Bojan Jerbić. 2005. - 2007., istraživač na tehnologijskom projektu ″Inteligentna stanica za posluživanje automatskog montažnog sustava″, TP 05/5046 projekt podržan od Hrvatskog instituta za tehnologiju, glavni istraživač prof. dr. sc. Bojan Jerbić. 2005. - 2007., istraživač na projektu ″Inteligentno automatsko sklapanje″, znanstveni projekt br. 0120015podržan od Ministarstava znanosti obrazovanja i športa RH, glavni istraživač prof. dr. sc. Božo Vranješ.
Znanstveno usavršavanje:	2009., Lipanj - Srpanj, istraživački posjet *KTH* Stockholm, Kraljevina Švedska.
Nagrade:	1998. - Davorin Bazjanac, FSB Zagreb, nagrada dodijeljena za izvrstan

	uspjeh u studiju. 1999. - Plaketa Fakulteta, nagrada dodijeljena za posebne uspjehe u studiju prigodom obilježavanja 80. godišnjice Dana Fakulteta. 2009. - Zlatno Teslino odličje dodijeljeno od Instituta Ruđer Bošković i časopisa Vidi u konkurenciji akademskih ustanova za projekt dvoručnog robotskog sustava. 2009. - Povelja za najbolji e-kolegij na Fakultetu strojarstva i brodogradnje dodijeljena prigodom održavanja 90. godišnjice Dana Fakulteta.

Biography

Name:	Petar Ćurković
E-mail:	petar.curkovic@fsb.hr
Date of birth:	02.12.1977.
Place of birth::	Zagreb, Hrvatska
Education:	2005. - 2010., Doctoral programme, Faculty of mechanical engineering and naval architecture, University of Zagreb 1996. - 2004., B.Sc. programme, Faculty of mechanical engineering and naval architecture, University of Zagreb 1992. - 1996., Secondary ,school Zagreb 1984. - 1992., Grammar school, Zagreb
Positions:	01.01.2005. - today, Faculty of mechanical engineering and naval architecture, University of Zagreb, Department of Robotics and Manufacturing systems automation 03. 04. 2004. - 31. 12. 2004., Yazaki Europe Ltd
Foregin languages:	Fluent command of german and english.
Sudjelovao na znanstveno-istraživačkim projektima:	2009. -, researcher ″ Improving GRAsping Movements by predictions based on Observation″, UKF project financed by Ministry of sciences, education and sports, *RH*, principal researcher prof. dr. sc. Danica Kragić. 2008. researcher, technological project ″ Automated assembly with dual arm robot system, TP-08/0120-53 project supported by Croatian Instutute of Technology, principal researcher prof. dr. sc. Bojan Jerbić. 2007. -, researcher ″Autonomous multiagent assembly″, scientific project no.. 120-1201948-1941 financed by Ministry of sciences, education and sports, *RH* principal researcher prof. dr. sc. Bojan Jerbić. 2005. - 2007., researcher, technological project ″Intelligent cell for feeding of automated assembly system″, TP 05/5046 project supported by Croatian Instutute of Technology, principal researcher prof. dr. sc. Bojan Jerbić. 2005. - 2007., researcher, scientific project ″Intelligent automated assembly″, no. 0120015 financed by Ministry of sciences, education and sports, *RH* principal researcher prof. dr. sc. Božo Vranješ.
Scientific development:	2009., June - July, research visit *KTH* Stockholm, Sweden.
Awards:	1998. - Davorin Bazjanac, FMENA Zagreb, excellent achievements during the study. 1999. – Faculty Plaque.

	2009. – Golden Tesla Medal, Ruđer Bošković Institute, awarded 1st prize for realized dual arm robotic setup. 2009. – Best e-lectures on FMENA, faculty of mechanical enginnering and naval architecture.