

UNIVERZA V MARIBORU

Fakulteta za elektrotehniko, računalništvo in informatiko

Doktorska disertacija

Uglaševanje šahovske ocenitvene funkcije s
pomočjo algoritma diferencialne evolucije

Maribor, april 2010

Borko Bošković



Univerza v Mariboru
Fakulteta za elektrotehniko,
računalništvo in informatiko

Doktorska disertacija

Uglaševanje šahovske ocenitvene funkcije s
pomočjo algoritma diferencialne evolucije

Avtor: Borko Bošković, univ. dipl. inž. rač. in inf.

Mentor: izr. prof. dr. Janez Brest

Somentor: red. prof. dr. Viljem Žumer

Avtor: Borko Bošković, univ. dipl. inž. rač. in inf.
Naslov: Uglješevanje šahovske ocenitvene funkcije s pomočjo algoritma diferencialne evolucije
UDK: 004.89.21(043.3)
Ključne besede: Uglješevanje šahovske ocenitvene funkcije, diferencialna evolucija, zgodovinski mehanizem, mehanizem nasprotij
Število izvodov: 9

Zahvala

Zahvaljujem se mentorju izr. prof. dr. Janezu Brestu in somentorju red. prof. dr. Viljemu Žumerju za strokovno pomoč in napore pri izdelavi doktorske disertacije. Zahvaljujem se tudi vsem sodelavcem laboratorija za računalniške arhitekture in jezike.

Še posebej se zahvaljujem ženi Sonji in staršem, ki so me podpirali in vzpodbujali pri mojem delu.

UDK: 004.89.21(043.3)

Ključne besede: Uглаševanje šahovske ocenitvene funkcije, diferencialna evolucija, zgodovinski mehanizem, mehanizem nasprotij

Uглаševanje šahovske ocenitvene funkcije s pomočjo algoritma diferencialne evolucije

Povzetek

V delu predstavljamo algoritem za uглаševanje šahovske ocenitvene funkcije, ki temelji na algoritmu diferencialne evolucije (DE). Ocenjevanje posameznikov v procesu uглаševanja vsebuje šum. Zato smo algoritmu DE dodali mehanizem nasprotij, ki izboljšuje učinkovitost algoritma DE. V algoritmu uглаševanja smo dodali še nov zgodovinski mehanizem, ki zmanjšuje šum pri ocenjevanju potencialno dobrih posameznikov. Ti posamezniki igrajo več iger z različnimi nasprotniki. Zgodovinski mehanizem uporablja dodatno populacijo, ki vsebuje potencialno dobre posameznike glede na zgodovino evolucijskega procesa. Ti posamezniki se s pomočjo zgodovinskega mehanizma vračajo v proces uглаševanja, čeprav so izumrli v določeni prejšnji generaciji. Tako načrtovan algoritem uглаševanja zmanjšuje šum pri ocenjevanju posameznikov, zmanjšuje možnost prekomernega učenja in posledično omogoča učinkovit proces uглаševanja.

UDK: 004.89.21(043.3)

Keywords: Chess evaluation function tuning, Differential evolution, History mechanism, Opposition-based optimization

Differential evolution for the Tuning of a Chess Evaluation Function

Abstract

We present a Differential Evolution (DE) based approach for chess evaluation function tuning. DE with opposition-based optimization is employed and upgraded with a history mechanism. The opposition-based optimization improves efficiency of DE because of noise in the evaluation of individuals. Additionally introduced history mechanism reduces noise, because potentially good individuals played more games through several generations. New history mechanism uses an auxiliary population containing potentially good individuals from history of the evolutionary process. This mechanism ensures that potentially good individuals remain within the evolutionary process, even though they died several generations back and later can be injected back into the evolutionary process. In such manner, developed algorithm reduces noise in the evaluation of individual, reduces the possibility of overfitting and consequently improves efficiency of whole tuning process.



UNIVERZA v MARIBORU

Slomškov trg 15, 2000 Maribor, Slovenija

Tel.: 02 23 55 280

Fax: 02 23 55 211

e-mail: rektorat@uni-mb.si

www: <http://www.uni-mb.si>

Maribor, 29. 1. 2010

Številka: DR 1/2010/425-MGM

Na osnovi 287., 140., 142. in 144. člena Statuta Univerze v Mariboru (Statut UM-UPB7, Ur. l. RS, št. 36/2009) ter sklepa 28. redne seje Senata Univerze v Mariboru z dne 29. 1. 2010 v zvezi z vlogo doktorskega kandidata Borka Boškovića za sprejem odločitve o predlagani temi doktorske disertacije in mentorja,

izdajam naslednji

SKLEP

Odobri se tema doktorske disertacije Borka Boškovića s Fakultete za elektrotehniko, računalništvo in informatiko z naslovom »Uglaševanje šahovske ocenitve funkcije s pomočjo algoritma diferencialne evolucije«. Kandidatu se za mentorja imenuje izr. prof. dr. Janez Brest, za somentorja pa red. prof. dr. Viljem Žumer. Kandidat mora članici predložiti izdelano doktorsko disertacijo v zadostnih izvodih najpozneje do 28. 1. 2014.

Obrazložitev:

Kandidat Borko Bošković je dne 22. 6. 2009 na Fakulteti za elektrotehniko, računalništvo in informatiko vložil vlogo za potrditev teme doktorske disertacije z naslovom »Uglaševanje šahovske ocenitve funkcije s pomočjo algoritma diferencialne evolucije«. Za mentorja je bil predlagan izr. prof. dr. Janez Brest, za somentorja pa red. prof. dr. Viljem Žumer.

Senat Fakultete za elektrotehniko, računalništvo in informatiko je na osnovi pozitivnega mnenja komisije za oceno teme doktorske disertacije, ki je ugotovila, da kandidat izpolnjuje pogoje za pridobitev doktorata znanosti, in ocenila, da je predlagana tema ustrezna, sprejel pozitivno mnenje in poslal predlog teme doktorske disertacije s predlogom mentorja in somentorja v odobritev Senatu univerze.

Senat Univerze v Mariboru je po proučitvi vloge in na osnovi določil Statuta Univerze v Mariboru sprejel svojo odločitev o predlagani temi doktorske disertacije in imenoval mentorja in somentorja, kot izhaja iz izreka.

V skladu s 144. členom Statuta Univerze v Mariboru mora kandidat za pridobitev doktorata znanosti najpozneje v štirih letih od dneva izdaje tega sklepa, članici

predložiti izdelano doktorsko disertacijo v zadostnih izvodih. Kandidatu je bil določen rok glede na datum sprejetja teme na pristojnem organu.

Pouk o pravnem sredstvu:

Zoper ta sklep je možna pritožba na Senat Univerze v Mariboru v roku 8 dni od prejema tega sklepa.



Rektor:
Prof. dr. Ivan Rozman

[Handwritten signature]

Obvestiti:

1. Kandidata.
2. Fakulteto.
3. Arhiv.

Kazalo

1	Uvod	1
2	Sorodna dela	6
2.1	Računalniški šah	6
2.2	Strojno učenje pri igranju iger	8
3	Algoritem uglaševanja	13
3.1	Diferencialna evolucija	13
3.1.1	Inicializacija	16
3.1.2	Mutacija	17
3.1.3	Križanje	17
3.1.4	Popravljanje	19
3.1.5	Selekcija	20
3.2	Mehanizem nasprotij	20
3.2.1	Mehanizem nasprotij pri inicializaciji	21
3.2.2	Mehanizem dinamičnih nasprotij	22
3.3	Algoritem uglaševanja	23
3.3.1	Inicializacija	25
3.3.2	Ocenjevanje posameznikov	26
3.3.3	Posodabljanje zgodovinske populacije	29
3.3.4	Selekcija	30
3.3.5	Ustvarjanje novih posameznikov	31
3.3.6	Zgodovinsko vbrizgavanje	33

4	Poskusi	35
4.1	Šahovski program	35
4.1.1	Ocenitvena funkcija šahovskega programa	36
4.2	Načrt poskusov	39
4.3	Uglaševanje brez ekspertnega znanja	40
4.4	Uglaševanje na ozkih intervalih	45
4.4.1	Vpliv šuma	48
4.4.2	Izboljšava in moč najbolj ocenjenega posameznika	51
4.4.3	Vpliv zgodovinskega mehanizma	58
4.5	Uglaševanje z evolucijskim algoritmom	62
4.5.1	Inicializacija	63
4.5.2	Mutacija	64
4.5.3	Ocenjevanje posameznikov	65
4.5.4	Selekcija	65
4.5.5	Rezultati	66
4.6	Uglaševanje na širokih intervalih	68
5	Zaključek	72

Slike

2.1	Področje ocenjevanja pozicij s pomočjo nevronske mreže.	11
4.1	Primer ocenjevanja pozicij.	37
4.2	Povprečne vrednosti parametrov	43
4.3	Standardne deviacije parametrov	44
4.4	Histogram zmag, remijev in porazov najbolje ocenjenega posameznika iz začetne populacije pri 7. zagonu.	51
4.5	Histogram zmag, remijev in porazov najbolje ocenjenega posameznika iz 7. zagona.	52
4.6	Bonferroni-Dunn test (referenčni algoritem: DE ^{10*})	61

Tabele

3.1	Strategije algoritma DE	15
4.1	Nastavitve eksperimenta pri uglaševanju materialnih vrednosti figur in mobilnosti	40
4.2	Povprečne vrednosti parametrov pri inicializaciji	41
4.3	Standardne deviacije parametrov pri inicializaciji	42
4.4	Povprečne vrednosti zadnje generacije	42
4.5	Standardne deviacije zadnje generacije	45
4.6	Nastavitve parametrov pri algoritmu, ki uporablja zgodovinski mehanizem	48
4.7	Rezultati uglaševanja (<i>igre</i> = 2)	49
4.8	Rezultati uglaševanja (<i>igre</i> = 5)	49
4.9	Rezultati uglaševanja (<i>igre</i> = 10)	50
4.10	Vpliv krmilnega parametra <i>igre</i>	50
4.11	Turnir – 5 minut za 40 potez.	53
4.12	Nastavitve algoritma, ki ni uporabljal zgodovinskega mehanizma	59
4.13	Rezultati uglaševanja brez uporabe zgodovinskega mehanizma	60
4.14	Vpliv zgodovinskega mehanizma	60
4.15	Rangiranje algoritmov s pomočjo Friedmanovega testa	61
4.16	Prilagojene <i>p</i> -vrednosti	62
4.17	Nastavitve evlucijskega algoritma	67
4.18	Rezultati uglaševanja evlucijskega algoritma	67
4.19	Primerjava evlucijskega algoritma z našim algoritmom	68
4.20	Rezultati uglaševanja na širših intervalih	68
4.21	Primerjava uglaševanja na ozkih in širokih intervalih	69
4.22	Ozki intervali uglaševanja.	70
4.23	Širši intervali uglaševanja.	71

5.1	Rezultati uglasovanja na ozkih intervalih (<i>igre</i> = 10)	82
5.2	Rezultati uglasovanja na ozkih intervalih (<i>igre</i> = 5)	85
5.3	Rezultati uglasovanja na ozkih intervalih (<i>igre</i> = 2)	88
5.4	Rezultati uglasovanja brez zgodovinskega mehanizma na ozkih intervalih	91
5.5	Rezultati uglasovanja z evolucijskim algoritmom	94
5.6	Rezultati uglasovanja na širših intervalih	97

Algoritmi

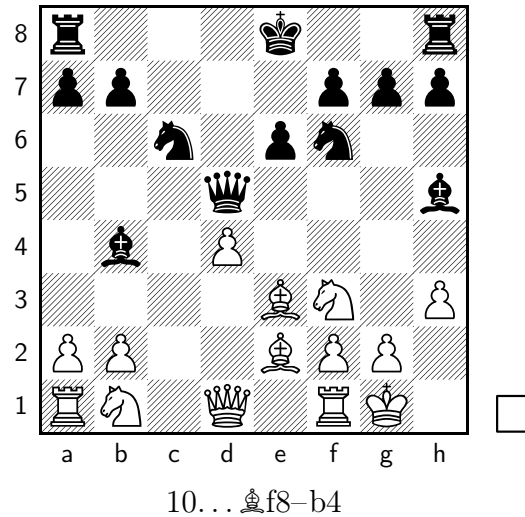
3.1	Algoritem diferencialne evolucije	16
3.2	Inicializacija v algoritmu DE	17
3.3	Mutacija v algoritmu DE	18
3.4	Križanje v algoritmu DE	18
3.5	Popravljanje v algoritmu DE	19
3.6	Selekcija v algoritmu DE	20
3.7	Algoritem DE, ki uporablja mehanizem nasprotij	21
3.8	Mehanizem nasprotij pri inicializaciji	22
3.9	Mehanizem dinamičnih nasprotij	23
3.10	Algoritem uglaševanja	25
3.11	Inicializacija v algoritmu uglaševanja	26
3.12	Ocenjevanje posameznikov	28
3.13	Posodabljanje zgodovinske populacije	30
3.14	Selekcija v algoritmu uglaševanja	31
3.15	Mutacija v algoritmu uglaševanja	32
3.16	Zgodovinsko vbrizgavanje	34
4.1	Evolucijski algoritem	64
4.2	Inicializacija v evolucijskem algoritmu	64
4.3	Mutacija v evolucijskem algoritmu	65
4.4	Ocenjevanje posameznikov	66

Simboli in oznake

DE	Diferencialna evolucija
FICS	Šahovski internetni strežnik (Free Internet Chess Server)
g	Trenutna generacija evolucijskega procesa
T_g	Trenutna populacija v g -ti generaciji
P_g	Poskusna populacija v g -ti generaciji
N_p	Velikost trenutne in poskusne populacije
$\vec{x}_i^{t,g}$	i -ti posameznik trenutne populacije v g -ti generaciji
$\vec{x}_i^{p,g}$	i -ti posameznik poskusne populacije v g -ti generaciji
$e_i^{t,g}$	Ocena i -tega posameznika v trenutni populaciji g -te generacije
$e_i^{p,g}$	Ocena i -tega posameznika v poskusni populaciji g -te generacije
D	Dimenzija problema, ki ga rešujemo in dimenzija vektorjev
$x_{i,j}^{t,g}$	j -ti elementi vektorja i -tega posameznika v trenutni populaciji g -te generacije oz. parametri ocenitvene funkcije
$x_{i,j}^{p,g}$	j -ti elementi vektorja i -tega posameznika v poskusni populaciji g -te generacije oz. parametri ocenitvene funkcije
\vec{x}_{min}	Vektor spodnjih mej intervalov
\vec{x}_{max}	Vektor zgornjih mej intervalov
C_r	Krmilni parameter križanja
F	Krmilni parameter mutacije
s	Strategija algoritma DE
GEN	Maksimalno število generacij evolucijskega procesa (zaustavitveni pogoj)
f	Funkcija oz. problem, ki ga rešujemo
λ	Drugi krmilni parameter mutacije

\vec{m}	Vektor mutacije
\vec{k}	Vektor križanja
GPL	GNU General Public License - licenca za prosto dostopno programsko opremo
ANSI	American National Standards Institute - Ameriški državni inštitut za standarde
GLib	Knjižnica, ki vsebuje programska orodja za različne platforme
UCI	Universal Chess Interface - vmesnik oz. protokol za komunikacijo med šahovskim programom in grafičnim vmesnikom
XBoard	vmesnik oz. protokol za komunikacijo med šahovskim programom in grafičnim vmesnikom
YBWC	Young Brothers Wait Concept - koncept za paralelizacijo šahovskih iskalnih algoritmov
ODE	Opposition-Based Differential Evolution - algoritem DE, ki temelji na nasprotjih
J_r	Jumping Rate - parameter, ki določa verjetnost dinamičnega nasprotja v generaciji
$x_{min,j}^g$	j -ti element vektorja spodnjih dinamičnih mej v generaciji g
$x_{max,j}^g$	j -ti element vektorja zgornjih dinamičnih mej v generaciji g
Z_g	Zgodovinska populacija
Z_{vel}	Število najboljših posameznikov v zgodovini
Z_{vbriz}	Število zgodovinskih vbrizgavanj
<i>globina</i>	Globina preiskovanja šahovskega programa
<i>igre</i>	Krmilni parameter, ki določa število odigranih iger pri ocenjevanju posameznikov
$i_i^{t,g}$	Število iger i -tega posameznika v trenutni populaciji g -te generacije
$i_i^{p,g}$	Število iger i -tega posameznika v poskusni populaciji g -te generacije
$r_i^{t,g}$	Dosežen rezultat i -tega posameznika v trenutni populaciji g -te generacije
$r_i^{p,g}$	Dosežen rezultat i -tega posameznika v poskusni populaciji g -te generacije
<i>gen_rez_i</i>	Razlika doseženih točk med i -tima posameznikoma iz trenutne in poskusne populacije
<i>starši</i>	Število staršev v populaciji

$potomci$	Število potomcev v populaciji
r_i	Število doseženih točk i -tega posameznika
\vec{s}_i^g	Vektor parametrov strategij
τ	Krmilni parameter mutacije
$N(\mu, \sigma)$	Gaussova naključna porazdelitev s povprečno vrednostjo μ in standardno deviacijo σ
ε	Spodnja meja parametrov strategij
$U(a, b)$	Uniformna naključna porazdelitev na intervalu $[a, b]$



Šahovska tabla je svet, figure so fenomen vesolja, pravila so čemur mi rečemo naravni zakoni in igralec na drugi strani nam je neviden. (Thomas Huxley)

1 Uvod

Računalniški šah ima že dolgo zgodovino raziskav na področju umetne inteligence. Šahovski programi so, z razvojem algoritmov in povečevanjem zmogljivosti strojne opreme, postali zelo močni šahovski igralci. To dejstvo dokazujejo dvoboji med računalniškimi sistemi in svetovnimi šahovskimi prvaki. Prvi dvoboj, v katerem je računalnik premagal svetovnega šahovskega prvaka, je bil odigran leta 1997. V dvoboju sta igrala računalnik Deep Blue in Garij Kasparov. Po tem dogodku so mnogi pričakovali, da se bo razvoj računalniškega šaha in šahovskih programov zaustavil. V nasprotju s temi pričakovanji se je ta razvoj nadaljeval. Leta 2005 je presenetil šahovski program Rybka, ki je postal najboljši šahovski program na vseh šahovskih lestvicah s prednostjo od 50 do 100 rating točk pred najbližjimi zasledovalci [27]. Tri leta kasneje je izšla nova različica tega programa z oznako Rybka 3. Igralna moč programa, glede na prejšnje različice, se je povečala od 50 do 100 rating točk.

Leta 2006 je šahovski program Deep Fritz 10, ki se je izvajal na osebni računalniku, premagal svetovnega šahovskega prvaka Vladimira Kramnika. Moč šahovskih programov na osebnih računalnikih je presegla moč svetovnega šahovskega prvaka. Zato se postavlja vprašanje, zakaj izboljševati že zelo močne šahovske programe? Odgovorov na to vprašanje je več. Profesionalni igralci šaha uporabljajo šahovske programe zato, da bi izboljšali svojo igralno moč. Šahovske programe uporabljamo tudi v dopisnem šahu in prostem slogu. V dopisnem šahu imajo igralci na voljo več časa, ki pa ga izkoristijo za različne analize. Pri analizah si pomagajo z računalniškimi programi, ki pozicije neutrudno analizirajo več ur ali celo dni. V prostem slogu so igralci časovno bolj omejeni, toda za razliko od klasičnega šaha si, prav tako kot v dopisnem šahu, pomagajo z analizami, kjer uporabljajo šahovske programe. Da bi ugotovili, kateri šahovski programi so boljši in kateri slabši, se med njimi igrajo turnirji, ki v zadnjem času postajajo vedno bolj zanimivi in popularni. Vse to kaže na dejstvo, da se računalniški šah še vedno razvija.

Pomemben del šahovskih programov, ki zelo vpliva na njihovo igralno moč, je ocenitvena funkcija (ang. *Evaluation Function*). Njena naloga je, da poda statično oceno pozicije oz. stanja igre. Zaradi kompleksnosti šahovske igre in velikega števila pozicij, ki jih šahovska igra vsebuje, ne obstaja preprosta ali eksaktna ocenitvena funkcija in verjetno tudi nikoli ne bo obstajala. Vseeno pa obstaja način za razvoj aproksimacijske ocenitvene funkcije, čeprav to opravilo ni enostavno [55]. Ocenitvena funkcija ponavadi vsebuje več aritmetičnih izrazov, katerih izračuni temeljijo na določenih parametrih. S pomočjo teh izrazov ocenitvena funkcija odkriva določena znanja v šahovskih pozicijah. Problem, ki se pojavi pri razvoju ocenitvene funkcije je, kako nastaviti vrednosti parametrov v aritmetičnih izrazih, ki so ponavadi med seboj odvisni. S pomočjo konvencionalnega razvoja programske opreme je to opravilo zelo težko izvedljivo zaradi časovne zahtevnosti. Razvijalec spreminja parametre in v fazi testiranja ugotavlja njihov vpliv na igralno moč programa ter ta postopek ponavlja določeno število iteracij. Ker je ta postopek časovno zahteven, je razvijalec omejen le na nekaj iteracij in tako zelo težko nastavi optimalne vrednosti parametrov. Član razvijalcev tima Deep Blue A. Joseph Hoane, Jr. je dejal [22]: “Pri razvoju se nam je pridružil starejši šahovski mojster Joel Benjamin, ki je igral proti računalniku Deep Blue. Tako smo odkrivali

napake in jih odpravljali vso leto. Ta postopek se na prvi pogled zdi neučinkovit, toda nismo našli boljšega načina za samodejno uglaševanje.” Iz tega je razvidno, da je razvoj šahovskih programov s pomočjo konvencionalnega pristopa časovno potraten, in da je samodejno uglaševanje težko opravilo.

Samodejno uglaševanje ali “učenje” je metoda, kjer razvijalca nadomesti računalnik. S tem se zmanjša časovna zahtevnost uglaševanja, ki pa je še vedno velika. Ko govorimo o samodejnih metodah uglaševanja pri računalniškem šahu, se moramo osredotočiti na algoritme, kot so učenje iz razlik v času (ang. *Temporal-Difference Learning*) [4, 5, 60, 62] in evolucijski algoritmi (ang. *Evolutionary Algorithms*) [30, 29, 24, 11, 12, 39, 45]. Ti pristopi usmerjajo proces uglaševanja glede na vmesne in končne rezultate dosežene v odigranih igrah.

V disertaciji se osredotočamo na samodejno uglaševanje z algoritmom diferencialne evolucije (DE) [47, 26, 57, 56]. To je preprost in učinkovit algoritem za globalno optimizacijo realno kodiranih problemov in uvršča se med evolucijske algoritme. Ta algoritem v disertaciji prilagodimo tako, da je primeren za uglaševanje parametrov šahovske ocenitvene funkcije. Šahovskim programom v času razvoja lahko parametre nastavi razvijalec skupaj s šahovskim ekspertom blizu optimalnih vrednosti. Pri uglaševanju takih programov, ki imajo relativno dobro nastavljene parametre, je zelo težko oceniti, kateri posamezniki (parametri ocenitvene funkcije) so boljši in kateri slabši. To omogoča, da slabši posamezniki dobijo boljšo oceno kot boljši posamezniki in preživijo v naslednje generacije. Do napačnega ocenjevanja posameznikov oz. šuma lahko pride v naslednjih primerih:

- slabši posameznik igra otvoritveno varianto, ki je neugodna za njegovega nasprotnika,
- učinek obzorja (ang. *Horizon Effect*) [35] v iskalnem algoritmu omogoči zmago slabšemu igralcu,
- boljši igralec ni toliko boljši, da bi v dvoboju, ki vsebuje malo število iger, dosegel boljši rezultat,

- posameznik ima take vrednosti parametrov, da mu omogočajo premagovanje določenih nasprotnikov, ki pa so v splošnem od njega boljši posamezniki in
- slabši posameznik igra proti nasprotnikom, ki so v povprečju slabši, in tako dobi boljšo oceno kot boljši posameznik.

Algoritem DE vsebuje populacijo posameznikov. Ti posamezniki igrajo določeno število iger, proti naključno izbranim posameznikom in na osnovi rezultatov iger dobijo ocene. Te ocene v algoritmu določajo, kateri posamezniki preživijo v naslednjo generacijo. Ker pa ocene posameznikov vsebujejo določene napake, se v procesu uglasovanja pojavi šum.

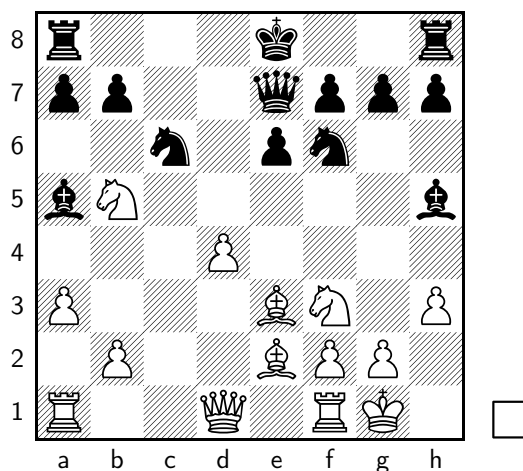
Da zmanjšamo šum in povečamo učinkovitost uglasovanja, je algoritmu DE dodan mehanizem nasprotij in zgodovinski mehanizem. Mehanizem nasprotij izboljšuje učinkovitost algoritma DE za probleme, ki vsebujejo šum. Ideja tega mehanizma je, da istočasno ovrednoti posameznike in njihove nasprotne posameznike. Nov mehanizem, ki smo ga dodali algoritmu DE in načrtovali tako, da obdrži potencialno dobre posameznike iz zgodovine evolucijskega procesa v dodatni populaciji in jih z določeno verjetnostjo vrača nazaj v evolucijski proces, smo poimenovali zgodovinski mehanizem. S pomočjo tega mehanizma potencialno dobri posamezniki igrajo več iger v različnih okoljih in proti različnim nasprotnikom. Posledica tega je manjši šum pri ocenjevanju potencialno dobrih posameznikov. Potencialno dobri posamezniki, ki so v določeni generaciji bili slabo ocenjeni in so "izumrli", se lahko vrnejo nazaj v proces uglasovanja. S tem se v procesu uglasovanja vnaša raznolikost in posledično tudi zmanjšuje možnost prekomernega učenja.

Zgodovinski mehanizem smo načrtovali po vzoru iz narave. Novi šahovski igralci, ki želijo postati dobri igralci, se morajo učiti s pomočjo analiz že odigranih iger. Eden od načinov za to je, da se učijo oz. analizirajo igre svetovnih šahovskih prvakov ali dobrih šahistov iz preteklosti. Tako smo po tem vzoru v evolucijski proces vpeljali zgodovinsko populacijo, ki vsebuje dobre igralce iz vseh prejšnjih generacij. S pomočjo vbrizgavanja zgodovinskih posameznikov nazaj v evolucijski proces vplivamo na ocenjevanje in ustvarjanje novih posameznikov oz. na celoten evolucijski proces.

Na osnovi opisanega problema in predstavljenega algoritma smo postavili naslednje hipoteze:

- **Hipoteza 1:** Algoritem diferencialne evolucije, ki uporablja mehanizma nasprotij, je primeren za uglaševanje šahovske ocenitvene funkcije.
- **Hipoteza 2:** Zgodovinski mehanizem izboljšuje učinkovitost algoritma diferencialne evolucije, ki uporablja mehanizem nasprotij.
- **Hipoteza 3:** Algoritem diferencialne evolucije, ki uporablja mehanizem nasprotij in zgodovinski mehanizem, omogoča učinkovito uglaševanje parametrov tako na širokih kot na ozkih intervalih uglaševanja.

Vsebina tega dela je organizirana na naslednji način. V drugem poglavju opišemo sorodna dela. Tretje poglavje vsebuje podroben opis uporabljenih mehanizmov in algoritem za uglaševanje šahovske ocenitvene funkcije. Četrto poglavje prikazuje opis izvedenih poskusov in dobljene rezultate. V petem poglavju je podan zaključek o opravljenem raziskovalnem delu in smernice za nadaljnje delo. Sledi še priloga, kjer so prikazani podrobnejši rezultati poskusov.



13... ♔d6-e7

Življenje je kot šahovska igra, spreminja se z vsako potezo. (kitajski pregovor)

2 Sorodna dela

V tem poglavju podajamo kratek pregled razvoja računalniškega šaha in strojnega učenja pri igranju iger. Pri strojnem učenju bomo podrobneje predstavili metode, ki so bile aplicirane na računalniškem šahu.

2.1 Računalniški šah

Leta 1769 je Madžar Baron Wolfgang von Kempelen zgradil šahovski stroj, ki je bil namenjen zabavi cesarice Marije Terezije [3]. Stroj je presenetljivo dobro igral igro šah. Razlog temu je bila goljufija. Znotraj stroja je bil skrit šahovski mojster, ki je namesto stroja igral šah.

Prve prave ideje o izgradnji šahovskega stroja, datirajo v obdobje matematika, fizika in računalnikarja Charlesa Babbage (1792-1871). Babbage je opisal analitični stroj oz. teoretično napravo, ki je bila predhodnik digitalnega računalnika. Za razliko od današnjih računalnikov ta naprava ni bila elektronska in ni bila nikoli zgrajena. Diferenčni

stroj oz. predhodnik analitičnega stroja je bil uspešno zgrajen leta 1991 [58]. Babbage je tudi spoznal, da je njegov analitični stroj v osnovi zmožen igrati šah.

Pionir računalniškega šaha je bil Shannon. Ukvarjal se je s problemom, kako izdelati računalniški program, katerega namen bo igranje šaha [55]. Pri reševanju tega problema je ugotovil, da mora šahovski program vsebovati ocenitveno funkcijo in iskalni algoritem. Ocenitvena funkcija podaja statično oceno pozicij oz. stanja v igri, zato ta vključuje ocenjevanje materialne prednosti, formacije kmetov, položaja figur, mobilnosti figur in zaščite kralja [42]. To pomeni, da ne obravnava posameznih potez, ampak ocenjuje pozicijo glede na položaj figur. V ta namen ocenitvena funkcija vsebuje aritmetične izraze, ki odkrivajo določena znanja in podajajo oceno, kateri od igralcev ima boljšo pozicijo. Za končno oceno pozicije in izbiro poteze za nadaljevanje igre je zadolžen iskalni algoritem. Da iskalni algoritem izbere potezo za nadaljevanje igre, mora preiskovati drevo igre in ocenjevati določene pozicije (vozlišča v drevesu) s pomočjo ocenitvene funkcije.

Zanimivo je dejstvo, da je prvi šahovski program zapisan pred iznajdbo računalnikov. Zapisal ga je vizionar Alan Turing, ki je predvideval, da bodo nekega dne iznašli računalnike, katere bo mogoče programirati in posledično z njimi tudi igrati šah. Svojega programa ni nikoli izvajal na računalniku. S pomočjo ročne simulacije je bil program zmožen igrati šah. Ocenitvena funkcija programa je vključevala parametre mobilnosti, zaščite figur, rokade, pozicije kmetov ter grožnje šaha in mata. Zabeležena je tudi igra [42] tega programa. V tej igri je program uporabljal globino iskanja 2, igral proti domnevno slabemu nasprotniku in igro izgubil.

Prvi dokumentiran delujoč šahovski program je bil implementiran leta 1956 (Los Alamos). Že prej pa so poročali o delujočem programu iz Sovjetske zveze, kar pa ni bilo potrjeno [42]. Nedolgo za tem so Bernstein in ostali predstavili šahovski program, katerega moč igranja je bila relativno mala. Program je preiskoval drevo igre do globine 4 [9]. Prvi program, ki je uporabljal alfa-beta algoritem, so predstavili Newell in ostali [46]. Program, ki je prvi premagal človeka v turnirskem igranju, je bil Machack VI [33]. Njegova igralna moč je znašala 1640 rating točk in pri igranju je

preiskoval drevo igre z globino 9 ali več.

Leta 1978 je šahovski program Chess 4.7 v turnirski igri premagal šahovskega prvaka Škotske (David Levy). Ken Thopson je zgradil računalnik, ki je vseboval nekaj 100 čipov. Ta računalnik se je imenoval Belle in je bil namenjen le igranju šaha. V času turnirske igre je dosegel mojstrsko kategorijo. Tako je bilo v 80. letih dosti truda usmerjenega v izgradnjo specifične strojne opreme, ki bi omogočala preiskovanje in ocenjevanje velikega števila pozicij. Nastal je tudi sistem Hitech, ki ga je razvil Berliner in je vseboval 64 procesorjev. Hsu je naredil še močnejši čip, ki ga je znal izkoriščati program Deep Thought. Ta program je hitro prekašal sistem Hitech, zmožen je bil preiskovati drevo igre do globine 10 in je postal prvi program, ki mu je uspelo premagati starejšega mednarodnega mojstra (Bent Larsen). Rating programa je bil ocenjen na 2745 točk. Tudi ta program je igral dvoboj proti škotskemu prvaku (David Levy). Levy je priznal, da je to prvič, da je igral proti programu, ki je bil boljši od njega [42] in rezultat dvoboja je bil 4:0.

Začetek zmag šahovskih programov nad svetovnimi prvaki se je začel maja 1997, ko je Garry Kasparov izgubil dvoboj proti računalniku Deep Blue (naslednik programa Deep Thought). Tudi programi, ki so se izvajali na osebnih računalnikih, so presegli moč svetovnega prvaka. Leta 2006 je program Deep Fritz 10, ki je tekkel na osebnem računalniku, v dvoboju premagal šahovskega prvaka Vladimira Kramnika. Program Rybka je leta 2005 zasedel prvo mesto na vseh rating lestvicah in ga zaseda še danes. To je tudi prvi program, katerega rating je presegel 3000 točk [27].

2.2 Strojno učenje pri igranju iger

Če razvijalci želijo izboljšati šahovski program, mora program najprej odigrati določeno število iger. Odigrane igre nato analiziramo in ugotavljamo, v katerih pozicijah je program izbiral slabe poteze. Sledi popravljanje algoritma in ocenitvene funkcije tako, da poskušamo odpraviti odkrite pomanjkljivosti. Pri tem je potrebno biti pozoren, da s popravki ne povzročimo druge napake v igranju programa. To opravilo predstavlja en cikel, in ga je potrebno večkrat ponoviti. Ker pa je opisano opravilo časovno zelo

zahtevno, smo omejeni le na nekaj ciklov, zato programa bistveno ne moremo izboljšati. To je tudi razlog, zakaj so raziskovalci poskušali najti metode strojnega učenja, ki bi izboljšale igralno moč šahovskih programov in nadomestile konvencionalni razvoj programske opreme. Podoben problem se pojavlja tudi pri razvoju drugih iger, kot so npr. go, dama, backgammon, shogi itd. Ta problem je raziskovalo veliko število raziskovalcev. Pregled raziskav na tem področju je podan v [38].

Eden od pionirjev na področju učenja programov, ki igrajo igre, je bil Arthur L. Samuel. Pokazal je, da lahko napišemo tak program, ki omogoča računalniku, da se uči igranja igre. To je pokazal na primeru igre dama. S pomočjo učenja je program dosegel večjo igralno moč glede na program, ki učenja ni uporabljal in ga je človek izboljševal s pomočjo konvencionalnega razvoja programske opreme [52, 53]. Učenje programa se je izvajalo samodejno in je trajalo le 8 do 10 ur. Programu so bila podana pravila igre, cilj, in seznam parametrov. Parametri niso imeli nastavljenih vrednosti in so vplivali na potek igre.

Strojno učenje so aplicirali tudi nad iskalnim algoritmom oz. za določanje zaporedja preiskovanih potez. Dobro zaporedje preiskovalnih potez omogoča algoritmu, da doseže večje globine preiskovanja oz. preiskuje daljša nadaljevanja v igri in tako tudi vpliva na igralno moč programa. Da bi izboljšali zaporedje potez, so uporabili učenje na osnovi ocenitvenih funkcij. Te funkcije so bile načrtovane tako, da so omogočale izbiranje in urejanje preiskovanih potez [43]. Podobno delo je bilo predstavljeno v [64], kjer pa so uporabili množico neenakostnih omejitev, ki so določale področje, v katerem ocenitvena funkcija najde pravo potezo.

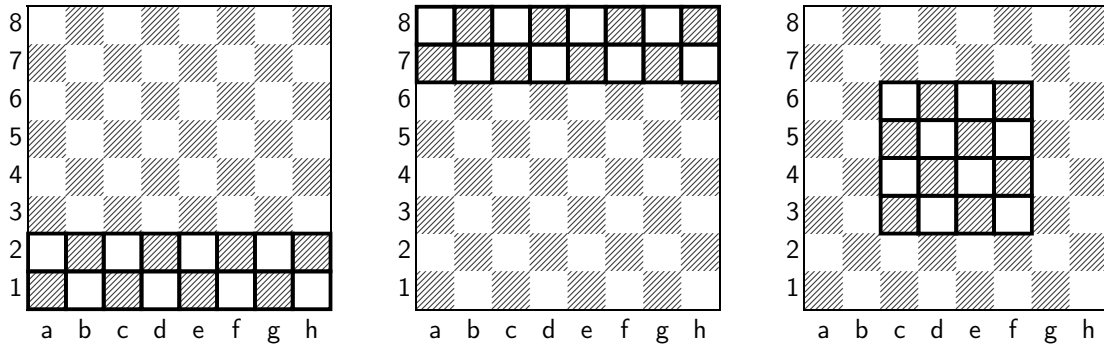
Zanimiva ideja je bila predstavljena v [41], kjer je bil uporabljen šahovski program Morph. Ta program je bil sposoben učenja s pomočjo vzorcev in drugih oblik učenja. NeuroChess je bil šahovski program, ki se je učil glede na rezultate odigranih iger [62]. Program je učil svojo ocenitveno funkcijo, ki je bila predstavljena v obliki nevronske mreže. Pri učenju je uporabljal več različnih metod strojnega učenja. Metoda učenje iz razlik v času za določanje materialnih vrednosti figur je bila predstavljena v [6]. To delo so kasneje razširili tako, da so dodatno določali še vrednosti v tabelah pozicijskih

vrednosti figur (ang. *Piece-Square Tables*) [8]. Podobno delo je predstavljeno v [7], kjer so s pomočjo učenja določali vrednosti figur za igro shogi (japonski šah).

Tim Deep Thought (kasneje Deep Blue) je uglaševal parametre ocenitvene funkcije s pomočjo iger, ki so bile odigrane na mojstrskem nivoju. Uglaševanje je temeljilo na maksimiranju ujemanja med potezo, izbrano s strani programa in šahovskega mojstra [1, 36, 2]. Da bi izboljšali ocenitveno funkcijo programa Deep Blue so jo uglaševali s pomočjo metode “comparison training”, kjer je program igral igre proti samemu sebi [61].

KnightCap je javno dostopen šahovski program, s pomočjo katerega so samodejno učili ocenitveno funkcijo oz. uglaševali njene parametre [4, 5]. Pri učenju je bila uporabljena metoda učenje iz razlik v času in sprotno učenje (ang. *on-line learning*) z igranjem na strežniku FICS (Free Internet Chess Server). Program se je uspešno učil in po 3 dneh igranja odigral 308 iger ter izboljšal svoj rating za 500 rating točk. Rating, ki ga je imel pred učenjem, je znašal 1650 rating točk, po učenju pa 2150 rating točk. V prispevku [32] so predstavili robustno metriko za določanje kakovosti ocenitvene funkcije. S pomočjo empirične metode gradientnega vzpenjanja in omenjene metrike so optimizirali utežene lastnosti, uporabljene v šahovski ocenitveni funkciji. Z dobljenimi utežmi je program dosegal podobno zmogljivost kot program, kjer so bile uteži nastavljene “ročno”.

V tem odstavku podajamo še pomembnejša dela na področju učenja igre v računalniških programih, ki niso zajeta v objavah klasičnega šaha. V prispevkih [17, 18] je avtor delal na statistični metodi za oblikovanje ocenitvene funkcije programa za igro reversi in na ogrodju za pol-avtomatično konstruiranje ocenitvenih funkcij. Ogrodje za učenje na osnovi primerjav in nevronska mreža, primerna za učenje ocenitvene funkcije igre backgammon, sta predstavljena v prispevku [59]. Za učenje ocenitvene funkcije v isti igri je bila uporabljena metoda učenje iz razlik v času, kjer so učili program TD-Gammon [60]. V prispevku [54] so pokazali, da je metoda učenje iz razlik v času dala podobne rezultate kot “ročno” uglaševanje uteži skozi več let v igri dama. Tudi evlucijski algoritmi so bili uspešno uporabljeni za določanje uteži nevronske mreže v



Slika 2.1: Področje ocenjevanja pozicij s pomočjo nevronske mreže.

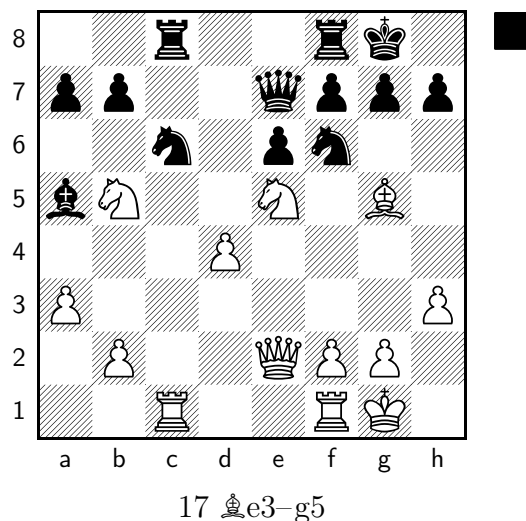
programih, ki so igrali igro dama [20, 21].

Za ugaševanje parametrov ocenitvenih funkcij so bili uporabljeni tudi principi evolucijskega računanja. Metoda za ugaševanje parametrov ocenitvene funkcije, ki temelji na evolucijskem algoritmu z uporabo iterativne selekcije, je predstavljena v [39]. V tem prispevku so pokazali, kako informacije o zmagi, porazu in remiju v povezavi s statistično analizo populacije uporabiti za določanje vrednosti parametrov ocenitvene funkcije. V delu, ki je predstavljeno v [45], je predlagan evolucijski algoritem z novo strategijo, kjer so ugaševali le najpomembnejše parametre ocenitvene funkcije. To strategijo so poimenovali “dynamic boundary strategy”. Uporabili so idejo, da so meje intervalov pri vsakem parametru ugaševanja dinamične oz. se spreminjajo skozi evolucijski proces. Parametre ocenitvene funkcije so ugaševali skozi 100 generacij in jo na koncu tudi uspešno uglasili. Tudi genetski algoritem je bil uporabljen za optimizacijo parametrov ocenitvene funkcije [44].

Odmevnejše delo strojnega učenja šahovskega programa, s pomočjo evolucijskega algoritma, je bilo predstavljeno v [30, 29, 24]. Evolucijski algoritem je bil uporabljen za ugaševanje parametrov ocenitvene funkcije. Ocenitvena funkcija je vsebovala material figur, tabele pozicijskih vrednosti in tri nevronske mreže. Ena nevronska mreža je ocenjevala sredino šahovske deske, preostali dve pa začetna položaja figur igralcev, kot prikazuje slika 2.1. Parametre so ugaševali v 10 neodvisnih zagonih skozi 50 generacij. Najboljše posameznike iz vsakega zagona so ocenili in izbrali najboljšega. Najboljši posameznik je dosegel igralno moč 2550 rating točk. Tako so z evolucijskim algoritmom in s pomočjo nevronske mreže izboljšali program za 371 rating točk. Podobno

delo je bilo opravljeno tudi za igro dama [20, 21, 28, 29].

Iz predstavljenih sorodnih del je razvidno, da so evolucijski algoritmi ena od možnih metod, ki jih lahko uporabimo za uglaševanje parametrov ocenitvene funkcije. Evolucijski algoritmi so bili že aplicirani na uglaševanje osnovnih parametrov ocenitvene funkcije in uglaševanje dodatnih podatkovnih struktur, kot so npr. nevronske mreže in tabele pozicijskih vrednosti. Tako bomo tudi mi poskušali zgraditi algoritem za uglaševanje parametrov ocenitvene funkcije, ki bo temeljil na algoritmu DE.



Noben tiran ne more matirati svobode. (Machiavelli)

3 Algoritem uglaševanja

V tem poglavju predstavljamo naš algoritem uglaševanja. Najprej opisujemo algoritem DE in mehanizem nasprotij, ki predstavljata osnovo našega algoritma, nato pa sledi še podrobna predstavitev našega algoritma.

3.1 Diferencialna evolucija

Algoritem diferencialne evolucije (ang. *Differential Evolution*) je enostaven in učinkovit algoritem za globalno optimizacijo. Predstavila sta ga Storn in Price [56, 57]. Od takrat je bil zaradi svoje enostavnosti in učinkovitosti uporabljen za reševanje različnih praktičnih problemov. Osnovna različica algoritma je bila modificirana in nastalo je več novih različic [47, 26, 14, 13, 15, 48, 49, 50, 23, 65]. Algoritem DE lahko opišemo kot preprost matematični model kompleksnega procesa naravne evolucije [26]. Tako se DE uvršča med algoritme umetne evolucije oz. evolucijske algoritme. Matematični model diferencialne evolucije je zasnovan na uporabi razlik med vektorji oz. posamezniki. Razlike med posamezniki se določajo s pomočjo hitrih in enostavnih aritmetičnih

operacij, ki ločujejo DE od ostalih metod za globalno optimizacijo.

DE je realno kodiran populacijski algoritem, ki v vsaki generaciji g vsebuje trenutno T_g in poskusno P_g populacijo. Trenutna populacija vsebuje posameznike, ki so preživeli prejšnjo generacijo, poskusna populacija pa posameznike, ki jih je ustvaril algoritem v trenutni generaciji in bodo tekmovali proti posameznikom iz trenutne populacije. Obe populaciji vsebujeta N_p posameznikov $(\vec{x}_i^{t,g}, \vec{x}_i^{p,g})$ in njihovih ocenitev $(e_i^{t,g}, e_i^{p,g})$. Posamezniki so predstavljeni v obliki D dimenzionalnih vektorjev. Elementi vektorjev $(x_{i,j}^{t,g}, x_{i,j}^{p,g})$ so realna števila, katerih vrednosti se nahajajo na določenih intervalih. Intervale parametrov določa uporabnik s pomočjo spodnjih \vec{x}_{min} in zgornjih \vec{x}_{max} mej. Opisano predstavitev algoritma simbolično prikazujejo neenačbi in enačbe (3.1).

$$\begin{aligned}
T_g &= \{\{\vec{x}_0^{t,g}, e_0^{t,g}\}, \{\vec{x}_1^{t,g}, e_1^{t,g}\}, \dots, \{\vec{x}_{N_p-1}^{t,g}, e_{N_p-1}^{t,g}\}\}, \\
P_g &= \{\{\vec{x}_0^{p,g}, e_0^{p,g}\}, \{\vec{x}_1^{p,g}, e_1^{p,g}\}, \dots, \{\vec{x}_{N_p-1}^{p,g}, e_{N_p-1}^{p,g}\}\}, \\
\vec{x}_i^{t,g} &= \{x_{i,0}^{t,g}, x_{i,1}^{t,g}, \dots, x_{i,D-1}^{t,g}\}, \quad \vec{x}_i^{p,g} = \{x_{i,0}^{p,g}, x_{i,1}^{p,g}, \dots, x_{i,D-1}^{p,g}\}, \\
\vec{x}_{min} &= \{x_{min,0}, x_{min,1}, \dots, x_{min,D-1}\}, \\
\vec{x}_{max} &= \{x_{max,0}, x_{max,1}, \dots, x_{max,D-1}\}, \\
x_{min,j} &\leq x_{i,j}^{t,g} \leq x_{max,j}, \\
x_{min,j} &\leq x_{i,j}^{p,g} \leq x_{max,j}, \\
x_{i,j}^{t,g}, x_{i,j}^{p,g}, e_i^{t,g}, e_i^{p,g}, x_{min,j}, x_{max,j} &\in \mathbb{R}, \\
i &\in \{0, N_p - 1\}, \quad j \in \{0, D - 1\}
\end{aligned} \tag{3.1}$$

Algoritem DE vsebuje nekaj parametrov, s pomočjo katerih lahko algoritem krmilimo oz. mu spreminjamo lastnosti in ga prilagajamo različnim problemom. Parametri so naslednji:

- F - parameter mutacije (ang. *Mutation Parameter*),
- C_r - parameter križanja (ang. *Crossover Parameter*),
- N_p - velikost populacije (ang. *Size of Population*),
- s - strategija algoritma DE (ang. *DE Strategy*),
- D - dimenzija problema (ang. *Dimension of Problem*),

- $x_{min,j}$ - spodnje meje parametrov uglaševanja (ang. *Low Boundaries*),
- $x_{max,j}$ - zgornje meje parametrov uglaševanja (ang. *High Boundaries*) in
- GEN - maksimalno število generacij oz. zaustavitveni pogoj (ang. *Stopping Condition*).

Parametri F , C_r , N_p , s in GEN predstavljajo krmilne parametre (ang. *Control Parameters*), in vplivajo na obnašanje algoritma. Parametre D , $x_{min,j}$ in $x_{max,j}$ določa problem, ki ga rešujemo. V tem poglavju predstavimo problem kot funkcijo f , za katero je potrebno poiskati globalni optimum. Glede na tip problema, lahko izberemo med več različnimi strategijami algoritma DE. Strategija določa način mutacije in križanja. Najbolj znane so prikazane v tabeli 3.1. Prvi stolpec predstavlja ime strategije, ki je sestavljeno tako, da pove za kakšno mutacijo in križanja gre. Drugi stolpec vsebuje način mutacije in tretji stolpec način križanja. V tabeli poleg že predstavljenih simbolov, zasledimo oznaki $\vec{x}_{best}^{t,g}$ in λ . Oznaka $\vec{x}_{best}^{t,g}$ predstavlja najboljšega posameznika v trenutni populaciji g -te generacije in oznaka λ predstavlja drugi krmilni parameter mutacije. V literaturi je podanih še več različnih strategij [14, 13, 49, 50], ki vpeljujejo različne koncepte v algoritem DE ter ga prilagajajo različnim problemom.

Tabela 3.1: Strategije algoritma DE

Strategija	Mutacija	Križanje
$DE/rand/1/bin$	$\vec{m} = \vec{x}_{r_1}^{t,g} + F \cdot (\vec{x}_{r_2}^{t,g} - \vec{x}_{r_3}^{t,g})$	bin
$DE/rand/2/bin$	$\vec{m} = \vec{x}_{r_1}^{t,g} + F \cdot (\vec{x}_{r_2}^{t,g} + \vec{x}_{r_3}^{t,g} - \vec{x}_{r_4}^{t,g} - \vec{x}_{r_5}^{t,g})$	bin
$DE/best/1/bin$	$\vec{m} = \vec{x}_{best}^{t,g} + F \cdot (\vec{x}_{r_1}^{t,g} - \vec{x}_{r_2}^{t,g})$	bin
$DE/best/2/bin$	$\vec{m} = \vec{x}_{best}^{t,g} + F \cdot (\vec{x}_{r_1}^{t,g} + \vec{x}_{r_2}^{t,g} - \vec{x}_{r_3}^{t,g} - \vec{x}_{r_4}^{t,g})$	bin
$DE/rand\ to\ best/1/bin$	$\vec{m} = \vec{x}_i^{t,g} + \lambda \cdot (\vec{x}_{best}^{t,g} - \vec{x}_{r_1}^{t,g}) + F \cdot (\vec{x}_{r_2}^{t,g} - \vec{x}_{r_3}^{t,g})$	bin
$DE/rand/1/exp$	$\vec{m} = \vec{x}_{r_1}^{t,g} + F \cdot (\vec{x}_{r_2}^{t,g} - \vec{x}_{r_3}^{t,g})$	exp
$DE/rand/2/exp$	$\vec{m} = \vec{x}_{r_1}^{t,g} + F \cdot (\vec{x}_{r_2}^{t,g} + \vec{x}_{r_3}^{t,g} - \vec{x}_{r_4}^{t,g} - \vec{x}_{r_5}^{t,g})$	exp
$DE/best/1/exp$	$\vec{m} = \vec{x}_{best}^{t,g} + F \cdot (\vec{x}_{r_1}^{t,g} - \vec{x}_{r_2}^{t,g})$	exp
$DE/best/2/exp$	$\vec{m} = \vec{x}_{best}^{t,g} + F \cdot (\vec{x}_{r_1}^{t,g} + \vec{x}_{r_2}^{t,g} - \vec{x}_{r_3}^{t,g} - \vec{x}_{r_4}^{t,g})$	exp
$DE/rand\ to\ best/1/exp$	$\vec{m} = \vec{x}_i^{t,g} + \lambda \cdot (\vec{x}_{best}^{t,g} - \vec{x}_{r_1}^{t,g}) + F \cdot (\vec{x}_{r_2}^{t,g} - \vec{x}_{r_3}^{t,g})$	exp

Algoritem 3.1 Algoritem diferencialne evolucije

Oznake: T_g : trenutna populacija, P_g : poskusna populacija, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, \vec{x}_{min} : vektor spodnjih mej parametrov, \vec{x}_{max} : vektor zgornjih mej parametrov, f : problem, ki ga rešujemo, g : številka trenutne generacije, GEN : maksimalno število generacij, i : indeks posameznika, \vec{m} : vektor mutacije, \vec{k} : vektor križanja, F : parameter mutacije, C_r : parameter križanja, $\vec{x}_i^{t,g}$: i -ti posameznik trenutne populacije in $\vec{x}_i^{p,g}$: i -ti posameznik poskusne populacije.

```

1: inicializacija( $T_0, N_p, D, \vec{x}_{min}, \vec{x}_{max}, f$ );
2: for  $g = 0$  to  $GEN$  do
3:   for  $i = 0$  to  $N_p$  do
4:      $\vec{m} = \text{mutacija}(T_g, i, F, N_p, D)$ ;
5:      $\vec{k} = \text{križanje}(\vec{x}_i^{t,g}, \vec{m}, C_r, D)$ ;
6:      $\vec{x}_i^{p,g} = \text{popravljanje}(\vec{k}, D, \vec{x}_{min}, \vec{x}_{max})$ ;
7:   end for
8:   selekcija( $T_g, P_g, N_p, D, f$ );
9: end for

```

V nadaljevanju poglavja predstavimo algoritma DE s pomočjo ene od najbolj razširjenih in uporabljenih strategij *DE/rand/1/bin*. Algoritem DE z opisano predstavitvijo in parametri prikazuje algoritem 3.1. Prikazan algoritem je preprost, vsebuje inicializacijo in dve programski zanki. Znotraj zank vsebuje naslednje operacije: mutacijo, križanje, popravljanje in selekcijo. Podrobnejši opis operacij sledi v nadaljevanju poglavja.

3.1.1 Inicializacija

Preden algoritem začne optimizacijo, inicializira vektorje in izračuna njihove ocenitve. Ker nima nobenega znanja o problemu, vrednosti vektorjem nastavi naključno, s pomočjo naključnega generatorja števil na definiranih intervalih ($\vec{x}_{min}, \vec{x}_{max}$). Način inicializacije prikazujeta enačbi (3.2) in algoritem 3.2. V enačbi je uporabljena spremenljivka $rand_{i,j}[0, 1)$, ki predstavlja naključno spremenljivko, katere zaloga vrednosti je v intervalu $[0, 1)$. V algoritmu je ta spremenljivka predstavljena v obliki funkcije $rand[0,1)$.

$$\begin{aligned}
 x_{i,j}^{t,0} &= x_{min,j} + (x_{max,j} - x_{min,j}) \cdot rand_{i,j}[0, 1) \\
 e_i^{t,0} &= f(x_i^{t,0})
 \end{aligned}
 \tag{3.2}$$

Algoritem 3.2 Inicializacija v algoritmu DE

Oznake: i : indeks posameznika, j : indeks elementa vektorja, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, g : številka trenutne generacije, $x_{i,j}^{t,g}$: j -ti element i -tega vektorja v trenutni populaciji, $x_{min,j}$: spodnja meja za j -te elemente vektorjev, $x_{max,j}$: zgornja meja za j -te elemente vektorjev, $\text{rand}[0,1)$: funkcija, ki vrne naključno vrednost na intervalu $[0,1)$, $e_i^{t,g}$: ocena i -tega posameznika v trenutni populaciji, $\vec{x}_i^{t,g}$: i -ti posameznik v trenutni populaciji in f : funkcija oz. problem, ki ga rešujemo.

```

1: for  $i = 0$  to  $N_p$  do
2:   for  $j = 0$  to  $D$  do
3:      $x_{i,j}^{t,g} = x_{min,j} + (x_{max,j} - x_{min,j}) \cdot \text{rand}[0,1)$ ;
4:   end for
5:    $e_i^{t,g} = f(\vec{x}_i^{t,g})$ ;
6: end for

```

3.1.2 Mutacija

Operacija mutacije izbere vektor iz trenutne populacije kot osnovni vektor (ang. *Base Vector*) $\vec{x}_{r_1}^{t,g}$ in ga premakne v določeno smer. Smer premikanja vektorja določa vektor razlike (ang. *Difference Vector*). Vektor razlike je določen z razliko med dvema naključno izbranimi vektorjema iz trenutne populacije $\vec{x}_{r_2}^{t,g} - \vec{x}_{r_3}^{t,g}$. Ta vektor se še skalira s pomočjo krmilnega parametra F . Ta parameter je ponavadi manjši od 1 in je definiran na intervalu $[0, 2]$. Tako skaliran vektor razlike se prišteje osnovnemu vektorju. Dobljen rezultat je mutiran vektor (ang. *Mutant Vector*) \vec{m} . Opisan način mutacije prikazujeta enačba (3.3) in algoritem 3.3. Oznake r_1, r_2 in r_3 predstavljajo indekse naključno izbranih posameznikov, ki so med seboj različni in hkrati različni od indeksa i .

$$\vec{m} = \vec{x}_{r_1}^{t,g} + F \cdot (\vec{x}_{r_2}^{t,g} - \vec{x}_{r_3}^{t,g}) \quad (3.3)$$

$$r_1, r_2, r_3 \in \{0, N_p - 1\}, \quad r_1 \neq r_2 \neq r_3 \neq i$$

3.1.3 Križanje

Iz tabel strategij vidimo, da ima DE dva tipa križanja: binarno in eksponentno. Tukaj bomo predstavili binarno križanje. To križanje za vsak element vektorja izračuna naključno vrednost $\text{rand}_{i,j} \in [0, 1)$. Če je ta vrednost večja od krmilnega parametra $C_r \in [0, 1]$, se elementu križanega vektorja (ang. *Crossover Vector*) k_j priredi element

Algoritem 3.3 Mutacija v algoritmu DE

Oznake: r_1, r_2, r_3 : naključno izbrani indeksi, i : indeks posameznika, j : indeks elementa vektorja, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, F : krmilni parameter mutacije, $\text{rand}\{0, N_p - 1\}$: funkcija, ki vrne naključno celo število na intervalu $[0, N_p - 1]$, m_j : j -ti element mutiranega vektorja in $x_{r_1,j}^{t,g}, x_{r_2,j}^{t,g}, x_{r_3,j}^{t,g}$: j -ti elementi vektorjev iz trenutne populacije, ki so na indeksih r_1, r_2 in r_3 .

```

1:  $r_1 = \text{rand}\{0, N_p - 1\}$ ;
2: while  $r_1 == i$  do
3:    $r_1 = \text{rand}\{0, N_p - 1\}$ ;
4: end while
5:  $r_2 = \text{rand}\{0, N_p - 1\}$ ;
6: while  $r_2 == i$  or  $r_2 == r_1$  do
7:    $r_2 = \text{rand}\{1, N_p - 1\}$ ;
8: end while
9:  $r_3 = \text{rand}\{0, N_p - 1\}$ ;
10: while  $r_3 == i$  or  $r_3 == r_1$  or  $r_3 == r_2$  do
11:    $r_3 = \text{rand}\{0, N_p - 1\}$ ;
12: end while
13: for  $j = 0$  to  $D$  do
14:    $m_j = x_{r_1,j}^{t,g} + F \cdot (x_{r_2,j}^{t,g} - x_{r_3,j}^{t,g})$ ;
15: end for

```

Algoritem 3.4 Križanje v algoritmu DE

Oznake: rand_j : naključni indeks, $\text{rand}\{0, D - 1\}$: funkcij, ki vrne naključno celo število na intervalu $[0, D - 1]$, D : dimenzija problema, i : indeks vektorja, j : indeks elementa vektorja, $\text{rand}_{i,j}$: naključna vrednost za j -ti element i -tega vektorja, $\text{rand}[0, 1)$: funkcija, ki vrne naključno vrednost na intervalu $[0, 1)$, C_r : krmilni parameter mutacije, k_j : j -ti element križanega vektorja, m_j : j -ti element mutiranega vektorja in $x_{i,j}^{t,g}$: j -ti element i -tega posameznika v trenutni populaciji g -te generacije.

```

1:  $\text{rand}_j = \text{rand}\{0, D - 1\}$ ;
2: for  $j = 0$  to  $D$  do
3:    $\text{rand}_{i,j} = \text{rand}[0, 1)$ 
4:   if  $\text{rand}_{i,j} < C_r$  or  $\text{rand}_j == j$  then
5:      $k_j = m_j$ ;
6:   else
7:      $k_j = x_{i,j}^{p,g}$ ;
8:   end if
9: end for

```

mutiranega vektorja m_j . V nasprotnem primeru element križanega vektorja dobi vrednost elementa vektorja iz trenutne populacije $x_{i,j}^{t,g}$. Pri operaciji križanja se uporablja

še spremenljivka $rand_j \in \{1, D\}$, ki je odgovorna zato, da križan vektor vsebuje vsaj en element mutiranega vektorja. Opisan način križanja prikazujeta enačba (3.4) in algoritem 3.4.

$$k_j = \begin{cases} m_j & \text{če } (rand_{i,j}[0, 1) < C_r) \vee (rand_j == j) \\ x_{i,j}^{t,g} & \text{drugače} \end{cases} \quad (3.4)$$

3.1.4 Popravljanje

Po operaciji križanja dobimo vektor križanja, katerega elementi k_j se lahko nahajajo izven določenih intervalov $[x_{min,j}, x_{max,j}]$. Zato je vektor križanja potrebno popraviti. Pri popravljanju imamo na voljo dve strategiji. Vrednosti, ki so izven intervalov, lahko popravimo tako, da jih nastavimo na bližjo mejo ali prezrcalimo nazaj v interval. Zrcaljenje vrednosti nazaj v interval prikazujeta enačba (3.5) in algoritem 3.5 za primer, ko je $F < 1$. Popravljen križan vektor je hkrati tudi poskusni vektor (ang. *Trial Vector*) $\bar{x}_i^{p,g} \in P_g$.

$$x_{i,j}^{p,g} = \begin{cases} x_{min,j} + (x_{min,j} - k_j) & k_j < x_{min,j} \\ x_{max,j} - (k_j - x_{max,j}) & k_j > x_{max,j} \\ k_j & \text{drugače} \end{cases} \quad (3.5)$$

Algoritem 3.5 Popravljanje v algoritmu DE

Oznake: i : indeks vektorja, j : indeks elementa vektorja, D : dimenzija problema, g : številka trenutne generacije, $x_{min,j}$: spodnja meja za j -te elemente vektorjev, $x_{max,j}$: zgornja meja za j -te elemente vektorjev, $x_{i,j}^{p,g}$: j -ti element i -tega posameznika v poskusni populaciji in k_j : j -ti element križanega vektorja.

```

1: for  $j = 0$  to  $D$  do
2:   if  $k_j < x_{min,j}$  then
3:      $x_{i,j}^{p,g} = x_{min,j} + (x_{min,j} - k_j)$ ;
4:   else
5:     if  $k_j > x_{max,j}$  then
6:        $x_{i,j}^{p,g} = x_{max,j} - (k_j - x_{min,j})$ ;
7:     else
8:        $x_{i,j}^{p,g} = k_j$ ;
9:     end if
10:  end if
11: end for

```

3.1.5 Selekcija

Selekcija določa, kateri posamezniki bodo preživeli v naslednjo generacijo. V ta namen se primerjata posameznika z enakim indeksom i iz trenutne in poskusne populacije ($\vec{x}_i^{p,g}$, $\vec{x}_i^{t,g}$). Odločitev o preživetju določa ocenitvena vrednost ($e_i^{t,g}$, $e_i^{p,g}$) (ang. *Fitness Value*) posameznikov. To vrednost pa določa funkcija oz. problem, ki ga rešujemo. V naslednjo generacijo preživi posameznik, ki ima boljšo ocenitev. Ko imata posameznika enako ocenitev v naslednjo generacijo preživi posameznik iz trenutne populacije. Primer selekcije za problem, kjer iščemo globalni maksimum, prikazujeta enačba (3.6) in algoritem 3.6.

$$e_i^{p,g} = f(\vec{x}_i^{p,g}),$$

$$\{\vec{x}_i^{t,g+1}, e_i^{t,g+1}\} = \begin{cases} \{\vec{x}_i^{t,g}, e_i^{t,g}\} & \text{če } e_i^{t,g} > e_i^{p,g} \\ \{\vec{x}_i^{p,g}, e_i^{p,g}\} & \text{drugače} \end{cases} \quad (3.6)$$

Algoritem 3.6 Selekcija v algoritmu DE

Oznake: i : indeks vektorja, g : številka trenutne generacije, N_p : velikost trenutne in poskusne populacije, $e_i^{p,g}$: ocenitev i -tega posameznika v poskusni populaciji, $e_i^{t,g}$: ocenitev i -tega posameznika v trenutni populaciji, f : problem, ki ga rešujemo, $\vec{x}_i^{t,g}$: i -ti posameznik v trenutni populaciji in $\vec{x}_i^{p,g}$: i -ti posameznik v poskusni populaciji.

```

1: for  $i = 0$  to  $N_p$  do
2:    $e_i^{p,g} = f(\vec{x}_i^{p,g}, D)$ 
3:   if  $e_i^{p,g} > e_i^{t,g}$  then
4:      $\vec{x}_i^{t,g} = \vec{x}_i^{p,g}$ ;
5:      $e_i^{t,g} = e_i^{p,g}$ ;
6:   end if
7: end for

```

3.2 Mehanizem nasprotij

Pri ocenjevanju posameznikov v procesu uglaševanja lahko pride do napačnih ocen oz. do šuma. Da bi izboljšali učinkovitost algoritma DE za naš problem, smo mu dodali mehanizem nasprotij. Mehanizem nasprotij temelji na učenju s pomočjo nasprotij (ang. *Opposition-Based Learning*) [63]. Ideja mehanizma je, da istočasno oceni določene posameznike in njihove "nasprotni" posameznike. Ta mehanizem s pomočjo selekcije

izbere posameznike, ki so verjetno bližje optimalni rešitvi in tako izboljša učinkovitost algoritma DE [49, 50]. Mehanizem nasprotij se znotraj algoritma DE uporablja pri inicializaciji, v obliki dinamičnih nasprotij pa v preostalem delu evolucijskega procesa, kot prikazuje algoritem 3.7. Vrstici 2 in 3 vključujeta mehanizem nasprotij pri inicializaciji, vrstice 10, 11 in 12 pa mehanizem dinamičnih nasprotij. Podrobnejši opis mehanizma sledi v naslednjih dveh podpoglavjih.

Algoritem 3.7 Algoritem DE, ki uporablja mehanizem nasprotij

Oznake: T_g : trenutna populacija, P_g : poskusna populacija, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, \vec{x}_{min} : vektor spodnjih mej parametrov, \vec{x}_{max} : vektor zgornjih mej parametrov, f : problem, ki ga rešujemo, g : številka trenutne generacije, GEN : maksimalno število generacij, i : indeks posameznika, \vec{m} : vektor mutacije, \vec{k} : vektor križanja, F : parameter mutacije, C_r : parameter križanja, J_r : parameter, ki določa verjetnost dinamičnega nasprotja v generaciji, $x_i^{t,g}$: i -ti posameznik trenutne populacije in $x_i^{p,g}$: i -ti posameznik poskusne populacije.

```

1: inicializacija( $T_0, N_p, D, \vec{x}_{min}, \vec{x}_{max}, f$ );
2:  $P_0 =$  nasprotje( $T_0, N_p, D, \vec{x}_{min}, \vec{x}_{max}$ );
3: selekcija( $T_0, P_0, N_p, D, f$ );
4: for  $g = 0$  to  $GEN$  do
5:   for  $i = 0$  to  $N_p$  do
6:      $\vec{m} =$  mutacija( $T_g, i, F, N_p, D$ );
7:      $\vec{k} =$  križanje( $\vec{x}_i^{t,g}, \vec{m}, C_r, D$ );
8:      $\vec{x}_i^{p,g} =$  popravljanje( $\vec{k}, D, \vec{x}_{min}, \vec{x}_{max}$ );
9:   end for
10:  if  $rand[0, 1] < J_r$  then
11:     $P_g =$  dinamičnoNasprotje( $P_g, N_p, D$ );
12:  end if
13:  selekcija( $T_g, P_g, N_p, D, f$ );
14: end for

```

3.2.1 Mehanizem nasprotij pri inicializaciji

Učinkovitost procesa uglaševanja je odvisna od oddaljenosti med rešitvijo in posamezniki inicializacije. Da bi oddaljenost zmanjšali in izboljšali učinkovitost algoritma, v poskusni populaciji ustvarimo posameznike, ki so nasproti posameznikom iz trenutne populacije. Nasprotne posameznike določajo intervali, na katerih uglašujemo parametre. Dobimo jih tako, da vrednosti parametrov prezrcalimo preko sredine intervalov.

Algoritem 3.8 Mehanizem nasprotij pri inicializaciji

Oznake: i : indeks posameznika, j : indeks elementa vektorja, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, $x_{min,j}$: j -ti element vektorja spodnjih mej, $x_{max,j}$: j -ti element vektorja zgornjih mej, $x_{i,j}^{p,0}$: j -ti element i -tega posameznika v poskusni populaciji pri inicializaciji in $x_{i,j}^{t,0}$: j -ti element i -tega posameznika v trenutni populaciji pri inicializaciji.

```

1: for  $i = 0$  to  $N_p$  do
2:   for  $j = 0$  to  $D$  do
3:      $x_{i,j}^{p,0} = x_{min,j} + x_{max,j} - x_{i,j}^{t,0}$ 
4:   end for
5: end for

```

Nato s pomočjo selekcije izberemo posameznike, ki preživijo v prvo generacijo. Ti posamezniki so verjetno bližje rešitvi, kot v primeru, če mehanizma nasprotij pri inicializaciji ne bi uporabili. Način delovanja opisanega mehanizma prikazujeta enačba (3.7) in algoritem 3.8.

$$x_{i,j}^{p,0} = x_{min,j} + x_{max,j} - x_{i,j}^{t,0} \quad (3.7)$$

3.2.2 Mehanizem dinamičnih nasprotij

Podobno kot pri inicializaciji se mehanizem nasprotij v nekoliko spremenjeni obliki uporablja skozi evlucijski proces. Pogostost uporabe mehanizma določa krmilni parameter J_r , kot prikazujejo vrstice 10, 11 in 12 v algoritmu 3.7. Ta mehanizem se razlikuje od mehanizma nasprotij pri inicializaciji, le po intervalih, glede na katere se določajo nasprotni posamezniki. Meje intervalov se dinamično spreminjajo skozi evlucijski proces, kot prikazujejo enačbe (3.8) in algoritem 3.9. Dinamične meje se določajo glede na posameznike iz trenutne populacije in jih označujemo kot $x_{min,j}^g, x_{max,j}^g$. Zaradi uporabe dinamičnih mej ta mehanizem imenujemo mehanizem dinamičnih nasprotij (ang. *Dynamic Opposition*).

$$\begin{aligned}
 x_{i,j}^{p,g} &= x_{min,j}^g + x_{max,j}^g - x_{i,j}^{t,g} \\
 x_{min,j}^g &= \min(x_{0,j}^{t,g}, x_{1,j}^{t,g}, \dots, x_{N_p-1,j}^{t,g}) \\
 x_{max,j}^g &= \max(x_{0,j}^{t,g}, x_{1,j}^{t,g}, \dots, x_{N_p-1,j}^{t,g})
 \end{aligned} \quad (3.8)$$

Algoritem 3.9 Mehanizem dinamičnih nasprotij

Oznake: i : indeks posameznika, j : indeks elementa vektorja, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, g : številka trenutne generacije, $x_{min,j}^g$: j -ti element vektorja spodnjih dinamičnih mej, $x_{max,j}^g$: j -ti element vektorja zgornjih dinamičnih mej, $x_{i,j}^{p,g}$: j -ti element i -tega posameznika iz poskusne populacije in $x_{i,j}^{t,g}$: j -ti element i -tega posameznika iz trenutne populacije.

```

1: for  $j = 0$  to  $D$  do
2:    $x_{min,j}^g = x_{max,j}^g = x_{0,j}^{t,g}$ 
3:   for  $i = 0$  to  $N_p$  do
4:     if  $x_{i,j}^{t,g} > x_{max,j}^g$  then
5:        $x_{max,j}^g = x_{i,j}^{t,g}$ 
6:     end if
7:     if  $x_{i,j}^{t,g} < x_{min,j}^g$  then
8:        $x_{min,j}^g = x_{i,j}^{t,g}$ 
9:     end if
10:  end for
11: end for
12: for  $i = 0$  to  $N_p$  do
13:   for  $j = 0$  to  $D$  do
14:      $x_{i,j}^{p,g} = x_{min,j}^g + x_{max,j}^g - x_{i,j}^{p,g}$ 
15:   end for
16: end for

```

3.3 Algoritem uglaševanja

Naš algoritem uglaševanja parametrov ocenitvene funkcije temelji na algoritmu DE in uporablja mehanizem nasprotij. V algoritmu smo spremenili predstavitev posameznikov in populacije glede na osnovni algoritem DE. V populaciji smo za vsakega posameznika dodali informaciji o številu odigranih iger in doseženem rezultatu. Ti dve informaciji potrebujemo za ocenjevanje posameznikov in njihovo primerjavo. Posamezniki pa so predstavljeni v obliki vektorjev, katerih elementi so parametri uglaševalne ocenitvene funkcije in so cela števila. Opisane spremembe prikazujejo neenačbi in enačbe (3.9).

$$\begin{aligned}
T_g &= \{\{\vec{x}_0^{t,g}, i_0^{t,g}, r_0^{t,g}\}, \{\vec{x}_1^{t,g}, i_1^{t,g}, r_1^{t,g}\}, \dots, \{\vec{x}_{N_p-1}^{t,g}, i_{N_p-1}^{t,g}, r_{N_p-1}^{t,g}\}\}, \\
P_g &= \{\{\vec{x}_0^{p,g}, i_0^{p,g}, r_0^{p,g}\}, \{\vec{x}_1^{p,g}, i_1^{p,g}, r_1^{p,g}\}, \dots, \{\vec{x}_{N_p-1}^{p,g}, i_{N_p-1}^{p,g}, r_{N_p-1}^{p,g}\}\}, \\
\vec{x}_i^{t,g} &= \{x_{i,0}^{t,g}, x_{i,1}^{t,g}, \dots, x_{i,D-1}^{t,g}\}, \quad \vec{x}_i^{p,g} = \{x_{i,0}^{p,g}, x_{i,1}^{p,g}, \dots, x_{i,D-1}^{p,g}\}, \\
\vec{x}_{min} &= \{x_{min,0}, x_{min,1}, \dots, x_{min,D-1}\}, \\
\vec{x}_{max} &= \{x_{max,0}, x_{max,1}, \dots, x_{max,D-1}\}, \\
x_{min,j} &\leq x_{i,j}^{t,g} \leq x_{max,j}, \\
x_{min,j} &\leq x_{i,j}^{p,g} \leq x_{max,j}, \\
x_{i,j}^{t,g}, x_{i,j}^{p,g}, i_i^{t,g}, i_i^{p,g}, r_i^{t,g}, r_i^{p,g}, x_{min,j}, x_{max,j} &\in \mathbb{Z}, \\
i &\in \{0, N_p - 1\}, \quad j \in \{0, D - 1\}
\end{aligned} \tag{3.9}$$

V procesu uglaševanja je potrebno posameznike oceniti. Na osnovi dobljene ocene se s pomočjo selekcije določi, kateri posamezniki bodo preživeli v naslednjo generacijo. Za ocenjevanje posameznikov smo v algoritem DE vključili igranje iger. Na osnovi doseženih rezultatov iger se določijo ocene posameznikov. Dobljene ocene se nato uporabijo v selekciji in določajo kateri posamezniki bodo preživeli v naslednjo generacijo.

Da bi izboljšali učinkovitost algoritma, smo v primeru uglaševanja šahovske ocenitvene funkcije, ki ima vrednosti parametrov nastavljene blizu optimalnih vrednosti, vpeljali zgodovinski mehanizem. S pomočjo tega mehanizma smo omogočili, da se potencialno dobri posamezniki, ki so v določeni generaciji dosegli dober rezultat in so kasneje "izumrli", vrnejo v evolucijski proces. Ti posamezniki odigrajo v evolucijskem procesu več iger proti različnim posameznikom, kar pomeni, da njihova ocena vsebuje manjšo napako. Zgodovinski mehanizem shranjuje dobre posameznike iz vsake generacije v zgodovinsko populacijo Z_g . Predstavitev zgodovinske populacije prikazuje enačba (3.10). Ta populacija vsebuje enako zgradbo kot trenutna in poskusna populacija, se pa od njih razlikuje po velikosti. Njena velikost se povečuje skozi evolucijski proces.

$$Z_g = \{\{\vec{x}_0^{z,g}, i_0^{z,g}, r_0^{z,g}\}, \{\vec{x}_1^{z,g}, i_1^{z,g}, r_1^{z,g}\}, \{\vec{x}_2^{z,g}, i_2^{z,g}, r_2^{z,g}\}, \dots\} \tag{3.10}$$

Celoten algoritem uglaševanja, ki vsebuje mehanizem nasprotij in zgodovinski mehanizem, prikazuje algoritem 3.10. Vpeljan zgodovinski mehanizem je razdeljen na dva dela. Prvi posodablja zgodovinsko populacijo oz. shranjuje potencialno dobre posameznike v zgodovinsko populacijo. Drugi del pa uporabi zgodovinsko populacijo in dobre posameznike vbrizguje nazaj v poskusno populacijo oz. v evolucijski proces.

Algoritem 3.10 Algoritem uglaševanja

Oznake: T_g : trenutna populacija, P_g : poskusna populacija, Z_g : zgodovinska populacija, Z_{vel} : število najboljših posameznikov v zgodovini, Z_{vbriz} : število zgodovinskih vbrizgavanj, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, \vec{x}_{min} : vektor spodnjih mej parametrov, \vec{x}_{max} : vektor zgornjih mej parametrov, f : problem, ki ga rešujemo, g : številka trenutne generacije, GEN : maksimalno število generacij, i : indeks posameznika, \vec{m} : vektor mutacije, \vec{k} : vektor križanja, F : parameter mutacije, C_r : parameter križanja, J_r : parameter, ki določa verjetnost dinamičnega nasprotja v generaciji, $globina$: globina preiskovanja šahovskega programa, $igre$: krmilni parameter, ki določa število odigranih iger pri ocenjevanju posameznikov, $x_i^{t,g}$: i -ti posameznik trenutne populacije in $x_i^{p,g}$: i -ti posameznik poskusne populacije.

```

1: inicializacija( $T_0, P_0, N_p, D, \vec{x}_{min}, \vec{x}_{max}$ );
2: oceniPosameznike( $T_0, P_0, N_p, D, globina, igre$ )
3: posodobiZgodovino( $Z_0, T_0, P_0, N_p, D$ );
4: selekcija( $T_0, P_0, N_p, D$ );
5: for  $g = 0$  to  $GEN$  do
6:   for  $i = 0$  to  $N_p$  do
7:      $\vec{m} = \text{mutacija}(T_g, i, F, N_p, D)$ ;
8:      $\vec{k} = \text{križanje}(\vec{x}_i^{t,g}, \vec{m}, C_r, D)$ ;
9:      $\vec{x}_i^{p,g} = \text{popravljanje}(\vec{k}, D, \vec{x}_{min}, \vec{x}_{max})$ ;
10:   end for
11:   if  $\text{rand}[0, 1] < J_r$  then
12:      $P_g = \text{dinamičnoNasprotje}(P_g, N_p, D)$ ;
13:   end if
14:    $P_g = \text{zgodVbrizgavanje}(P_g, Z_g, Z_{vel}, Z_{vbriz}, N_p, D)$ ;
15:   oceniPosameznike( $T_g, P_g, N_p, D, globina, igre$ )
16:   posodobiZgodovino( $Z_g, T_g, P_g, N_p, D$ );
17:   selekcija( $T_g, P_g, N_p, D$ );
18: end for

```

3.3.1 Inicializacija

Pri inicializaciji se vrednosti posameznikom trenutne populacije določijo naključno na definiranih intervalih. Vrednosti posameznikom poskusne populacije določi mehanizem nasprotij. Spremenljivke za ocenjevanje posameznikov, število odigranih iger in doseženi rezultati se inicializirajo na 0. Opisano inicializacijo prikazujejo enačbe (3.11) in algoritem 3.11.

Algoritem 3.11 Inicializacija v algoritmu uglaševanja

Oznake: i : indeks posameznika, j : indeks elementa vektorja, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, $x_{i,j}^{t,0}$: j -ti element i -tega vektorja v trenutni populaciji, $x_{i,j}^{p,0}$: j -ti element i -tega vektorja v poskusni populaciji, $x_{min,j}$: spodnja meja za j -te elemente vektorjev, $x_{max,j}$: zgornja meja za j -te elemente vektorjev in $\text{rand}[0,1)$: funkcija, ki vrne naključno vrednost na intervalu $[0,1)$.

```

1: for  $i = 0$  to  $N_p$  do
2:   for  $j = 0$  to  $D$  do
3:      $x_{i,j}^{t,0} = x_{min,j} + (x_{max,j} - x_{min,j}) \cdot \text{rand}[0,1)$ ;
4:      $x_{i,j}^{p,0} = x_{min,j} + x_{max,j} - x_{i,j}^{t,0}$ 
5:   end for
6:    $i_i^{t,0} = i_i^{p,0} = 0$ ;
7:    $r_i^{t,0} = r_i^{p,0} = 0$ ;
8: end for

```

$$\begin{aligned}
 x_{i,j}^{t,0} &= x_{min,j} + (x_{max,j} - x_{min,j}) \cdot \text{rand}_{i,j}[0,1) \\
 x_{i,j}^{p,0} &= x_{min,j} + x_{max,j} - x_{i,j}^{t,0} \\
 i_i^{t,0} &= i_i^{p,0} = r_i^{t,0} = r_i^{p,0} = 0
 \end{aligned}
 \tag{3.11}$$

3.3.2 Ocenjevanje posameznikov

Da bi ocenili posameznike, morajo ti oz. šahovski program odigrati igre. Igranje iger je lahko organizirano na različne načine. V našem algoritmu vsak posameznik iz trenutne in poskusne populacije odigra določeno število iger (parameter *igre*), polovico z belimi in polovico s črnimi figurami, proti naključno izbranim posameznikom iz trenutne populacije. Opisan način igranja iger in ocenjevanja posameznikov prikazujeta enačbi (3.12) in algoritem 3.12. V algoritmu je uporabljena spremenljivka *gen_rezi*, ki predstavlja razliko rezultatov med i -tima posameznikoma iz trenutne in poskusne populacije. Ta spremenljivka je vpeljana, da omogoči selekciji primerjavo i -tih posameznikov trenutne in poskusne populacije (podpoglavje 3.3.4). Pri ocenjevanju posameznikov, se posodablja še število odigranih iger in doseženi rezultati. Ti podatki se kasneje uporabljajo znotraj zgodovinskega mehanizma. V algoritmu je prikazan še krmilni parameter *igre*, ki določa število odigranih iger v procesu ocenjevanje posameznikov.

$$\begin{aligned}
gen_rez_i &= \left[\sum_{k=0}^{igre/2-1} \text{igrajIgro}(\vec{x}_i^{t,g}, \vec{x}_{m_{i,k}}^{t,g}, globina) + \right. \\
&\quad \left. \sum_{k=igre/2}^{igre-1} \left(1 - \text{igrajIgro}(\vec{x}_{m_{i,k}}^{t,g}, \vec{x}_i^{t,g}, globina) \right) \right] - \quad (3.12) \\
&\quad \left[\sum_{k=0}^{igre/2-1} \text{igrajIgro}(\vec{x}_i^{t,g}, \vec{x}_{n_{i,k}}^{t,g}, globina) + \right. \\
&\quad \left. \sum_{k=igre/2}^{igre-1} \left(1 - \text{igrajIgro}(\vec{x}_{n_{i,k}}^{t,g}, \vec{x}_i^{t,g}, globina) \right) \right] \\
m_{i,k}, n_{i,k} &= \text{rand}\{0, N_p - 1\}
\end{aligned}$$

Pri ocenjevanju posamezniki igrajo z določeno globino preiskovanja. Globina določa kako veliko drevo bo posameznik preiskoval. Večja globina določa večje drevo in posledično več potrebnega časa, da posameznik izbere potezo. Na drugi strani pa večja globina pomeni tudi, da je izbrana poteza verjetno boljša. V otvoritveni fazi igre posamezniki uporabljajo otvoritveno knjižnico, v fazi končnice pa baze končnic. Otvoritvena knjižnica za pozicije, ki jih vsebuje, ponuja več različnih nadaljevanj oz. potez. Posamezniki iz nabora ponujenih potez naključno izberejo eno. Tak način igranja otvoritev omogoča uglaševanje parametrov glede na otvoritve v otvoritveni knjižnici. Brez uporabe otvoritvene knjižnice bi se posamezniki uglasili le na osnovi potez, ki jih vrne iskalni algoritem. Rezultat iskalnega algoritma je odvisen od ocenitvene funkcije oz. od njenih parametrov. Glede na to, da smo naš algoritem uglaševanja zasnovali na malem številu posameznikov, ki hitro konvergira proti rešitvam, bi se posamezniki brez uporabe otvoritvene knjižnice uglasili le na nekaj otvoritvenih variant. Podobno kot v otvoritveni fazi igre, posamezniki v končni fazi igre uporabljajo dodatne informacije. V končnicah oz. pozicijah, ki vsebujejo 5 ali manj figur, posamezniki uporabljajo baze končnic in igrajo brez napak. Opisan način igranja iger, s pomočjo otvoritvene knjižnice in baz končnic, omogoča uglaševanje parametrov, ki se večinoma uporabljajo za ocenjevanje pozicij sredine igre.

Posameznik v igri zmagaja, ko doseže zmagovito pozicijo iz baz končnic, ko je njegova ocenitev pozicije večja od 400 točk in nasprotnikova manjša od -400 točk (prednost 400

Algoritem 3.12 Ocenjevanje posameznikov

Oznake: N_p : velikost trenutne in poskusne populacije, gen_rez_i : razlika doseženih točk med i -tima posameznikoma iz trenutne in poskusne populacije, $igre$: krmilni parameter, ki določa število odigranih iger, m : nasprotnikov indeks, $rand\{0, N_p - 1\}$: generator naključnih števil iz množice $\{0, N_p - 1\}$, $rezultat$: dosežen rezultat v odigrani igri, $globina$: globina preiskovanja šahovskega programa, $\bar{x}_i^{t,g}, \bar{x}_m^{t,g}$: posameznika iz trenutne populacije, $\bar{x}_i^{p,g}$: posameznik iz poskusne populacije, g : številka trenutne generacije, $r_i^{t,g}, r_m^{t,g}, r_i^{p,g}$: doseženi rezultati posameznikov, $i_i^{t,g}, i_m^{t,g}, i_i^{p,g}$: število odigranih iger posameznikov.

```

1: for  $i = 0$  to  $N_p$  do
2:    $gen\_rez_i = 0$ ;
3:   {Ocenjevanje posameznikov trenutne populacije}
4:   for  $l = 0$  to  $igre$  do
5:      $m = i$ ;
6:     while  $m == i$  do
7:        $m = rand\{0, N_p - 1\}$ ;
8:     end while
9:     if  $l \% 2 == 0$  then
10:       $rezultat = igrajIgro(\bar{x}_i^{t,g}, \bar{x}_m^{t,g}, globina)$ ;
11:    else
12:       $rezultat = 1 - igrajIgro(\bar{x}_m^{t,g}, \bar{x}_i^{t,g}, globina)$ ;
13:    end if
14:     $gen\_rez_i = gen\_rez_i + rezultat$ ;
15:     $r_i^{t,g} = r_i^{t,g} + rezultat$ ;  $r_m^{t,g} = t_m^{t,g} + 1 - rezultat$ ;
16:     $i_i^{t,g} = i_i^{t,g} + 1$ ;  $i_m^{t,g} = i_m^{t,g} + 1$ ;
17:  end for
18:  {Ocenjevanje posameznikov poskusne populacije}
19:  for  $l = 0$  to  $igre$  do
20:     $m = i$ ;
21:    while  $m == i$  do
22:       $m = rand\{0, N_p - 1\}$ ;
23:    end while
24:    if  $l \% 2 == 0$  then
25:       $rezultat = igrajIgro(\bar{x}_i^{p,g}, \bar{x}_m^{t,g}, globina)$ ;
26:    else
27:       $rezultat = 1 - igrajIgro(\bar{x}_m^{t,g}, \bar{x}_i^{p,g}, globina)$ ;
28:    end if
29:     $gen\_rez_i = gen\_rez_i - rezultat$ ;
30:     $r_i^{p,g} = r_i^{p,g} + rezultat$ ;  $r_m^{t,g} = t_m^{t,g} + 1 - rezultat$ ;
31:     $i_i^{p,g} = i_i^{p,g} + 1$ ;  $i_m^{t,g} = i_m^{t,g} + 1$ ;
32:  end for
33: end for

```

točk predstavlja prednost štirih kmetov in zmago z veliko verjetnostjo) ali v primeru, ko matira nasprotnika. Igra je končana z remijem, ko se doseže remi pozicija iz baze končnic, ko se v igri pozicija ponovi tri krat ali v primeru pravila 50 potez (v 50 potezah ni prišlo do jemanja figur ali premikanja kmetov). Igre smo dodatno omejili na 150 potez, tako da se končajo z remijem, ko dosežejo to mejo. Posameznik izgubi igro, ko njegov nasprotnik zmaga. Z igranjem iger posamezniki zbirajo točke. Za zmago dobi posameznik 1 točko, za remi $\frac{1}{2}$ točke in 0 točk za poraz. To so tudi vrednosti, ki jih vrača funkcija *igrajIgro*, ki je prikazana v enačbi (3.12) in algoritmu 3.12. Predstavljen način igranja iger se konča hitreje in s tem pohitri proces ocenjevanja, kakor tudi celoten evolucijski proces oz. uglaševanje parametrov. Da bi ocenjevanje še dodatno pohitrili, smo algoritem ocenjevanja paralelizirali. Tako lahko učinkovito izkoristimo več-jedrne procesorje, kakor tudi večprocesorske računalnike.

Opisani način ocenjevanja posameznikov omogoča, da slabši posameznik dobi boljšo oceno kot boljši posameznik zaradi narave šahovske igre. Primere, ko se to lahko zgodi, smo predstavili v poglavju 1. Zaradi možnosti napačne ocene posameznika se v algoritmu uglaševanja pojavi šum. Šum lahko zmanjšamo s številom odigranih iger. Večje, kot je število odigranih iger, manjša je verjetnost, da bomo dobili napačno oceno. Na drugi strani pa, večje kot je število odigranih iger, večja je časovna zahtevnost algoritma. Da bi zmanjšali vpliv šuma na proces uglaševanja pri malem številu igranja iger, smo v algoritmu vpeljali mehanizem zgodovine. Ta mehanizem posameznike, ki so dobro ocenjeni in predstavljajo potencialno dobre posameznike, obdrži v "igri". Dobro ocenjeni posamezniki se shranjujejo v zgodovinsko populacijo in s pomočjo mehanizmov zgodovine, vplivajo na evolucijski proces, čeprav so v evolucijskem procesu že "izumrli".

3.3.3 Posodabljanje zgodovinske populacije

Posodabljanje zgodovinske populacije je prvi del zgodovinskega mehanizma in je v algoritmu 3.10 prikazan v vrsticah 3 in 16. Po ocenjevanju je potrebno v zgodovinsko populacijo dodati nove posameznike in posodobiti podatke o obstoječih posameznikih. Način posodabljanja zgodovinske populacije prikazuje algoritem 3.13. V zgodovinsko populacijo se dodajajo le posamezniki iz trenutne populacije. Ti posamezniki so preživeli vsaj eno generacijo in so odigrali več iger proti različnim posameznikom. Ocena

teh posameznikov ima verjetno manjšo napako kot ocena posameznikov iz poskusne populacije. Pri dodajanju posameznika se dodata še podatka o številu odigranih iger in doseženem rezultatu. V primeru posodabljanja se za določenega posameznika ta dva podatka posodobita. Pri posodabljanju sodelujejo poleg posameznikov iz trenutne populacije tudi posamezniki iz poskusne populacije. Razlog temu je zgodovinsko vbrizgavanje (podpoglavje 3.3.6), ki dobre posameznike iz zgodovine vbrizguje v poskusno populacijo.

Algoritem 3.13 Posodabljanje zgodovinske populacije

Oznake: N_p : velikost trenutne in poskusne populacije, Z_g : zgodovinska populacija, g : številka trenutne generacije, $\vec{x}_i^{t,g}$: i -ti posameznik iz trenutne populacije, $\vec{x}_i^{p,g}$: i -ti posameznik iz poskusne populacije, $r_i^{t,g}, r_i^{p,g}$: doseženi rezultati posameznikov, $i_i^{t,g}, i_i^{p,g}$: število odigranih iger posameznikov.

```

1: for  $i = 0$  to  $N_p$  do
2:   if populacijaVsebuje( $Z_g, \vec{x}_i^{t,g}$ ) then
3:     posodobiPopulacijo( $Z_g, \vec{x}_i^{t,g}, i_i^{t,g}, r_i^{t,g}$ );
4:   else
5:     dodajVPopulacijo( $Z_g, \vec{x}_i^{p,g}, i_i^{p,g}, r_i^{p,g}$ );
6:   end if
7:   if populacijaVsebuje( $Z_g, \vec{x}_i^{p,g}$ ) then
8:     posodobiPopulacijo( $Z_g, \vec{x}_i^{p,g}, i_i^{p,g}, r_i^{p,g}$ );
9:   end if
10: end for
  
```

3.3.4 Selekcija

Selekcija je v algoritmu uglaševanja zelo podobna selekciji osnovnega algoritma DE. Sprememba selekcije je posledica načina ocenjevanja posameznikov. Pri ocenjevanju smo vpeljali za vsak par posameznikov spremenljivko gen_rez_i . Ta spremenljivka določa razliko doseženih rezultatov med i -tima posameznikoma iz trenutne in poskusne populacije v trenutni generaciji. S pomočjo te spremenljivke selekcija določa, kateri posamezniki bodo preživeli v naslednjo generacijo. Način delovanja selekcije prikazujeta enačba (3.13) in algoritem 3.14.

$$\{\vec{x}_i^{t,g+1}, i_i^{t,g+1}, r_i^{t,g+1}\} = \begin{cases} \{\vec{x}_i^{t,g}, i_i^{t,g}, r_i^{t,g}\} & gen_rez_i \geq 0 \\ \{\vec{x}_i^{p,g}, i_i^{p,g}, r_i^{p,g}\} & \text{drugače} \end{cases} \quad (3.13)$$

Algoritem 3.14 Selekcija v algoritmu uglaševanja

Oznake: i : indeks vektorja, g : številka trenutne generacije, N_p : velikost trenutne in poskusne populacije, $i_i^{p,g}$: število odigranih iger i -tega posameznika v poskusni populaciji, $i_i^{t,g}$: število odigranih iger i -tega posameznika v trenutni populaciji, $r_i^{p,g}$: dosežen rezultat i -tega posameznika v poskusni populaciji, $r_i^{t,g}$: dosežen rezultat i -tega posameznika v trenutni populaciji, $\bar{x}_i^{t,g}$: i -ti posameznik v trenutni populaciji in $\bar{x}_i^{p,g}$: i -ti posameznik v poskusni populaciji.

```

1: for  $i = 0$  to  $N_p$  do
2:   if  $gen\_rez_i < 0$  then
3:      $\bar{x}_i^{t,g} = \bar{x}_i^{p,g}$ ;
4:      $i_i^{t,g} = i_i^{p,g}$ ;
5:      $r_i^{t,g} = r_i^{p,g}$ ;
6:   end if
7: end for

```

V primeru, ko i -ta posameznika dosežeta enak rezultat oz. je vrednost spremenljivke gen_rez_i enaka 0, v naslednjo generacijo preživi posameznik iz trenutne populacije. Ta posameznik je verjetno preživel več generacij, odigral več iger proti različnim nasprotnikom in verjetno vsebuje boljše parametre. Selekcija poleg posameznika v naslednjo generacijo prenaša tudi podatka o odigranih igrah in doseženem rezultatu.

Prikazana selekcija določa, kateri posamezniki bodo preživeli v naslednjo generacijo glede na učinkovitost v trenutni generaciji. Če bi uporabili učinkovitost posameznikov glede na vse igre, ki so bile odigrane v evolucijskem procesu, bi posamezniki z dobro učinkovitostjo iz trenutne populacije lahko "izumrli". To bi se zgodilo, ko bi določen posameznik igral proti slabšim posameznikom in preživel nekaj generacij. V novi generaciji, kjer bi tekmoval z boljšim posameznikom, bi dosegel slabšo učinkovitost. Zaradi dobrih rezultatov iz prejšnjih generacij bi preživel v naslednjo generacijo. Iz tega razloga smo pri selekciji uporabili učinkovitost posameznikov iz trenutne generacije. Tako imajo posamezniki enako možnost za preživetje, ne glede na to, kako učinkoviti so bili v prejšnjih generacijah.

3.3.5 Ustvarjanje novih posameznikov

Posamezniki se ustvarjajo s pomočjo mutacije, križanja in mehanizma dinamičnega nasprotja. Kot smo že pokazali v tabeli 3.1, algoritem DE vsebuje več različnih strategij. Glede na naravo problema, ki ga rešujemo, smo v algoritmu uporabili novo strategijo

$DE/best_z/1$. Ta strategija je zelo podobna znani strategiji $DE/best/1$, ki je primerna za male populacije in hitro konvergenco [26]. Razlika med omenjenima strategijama je le v izbiri najboljšega posameznika. Strategija $DE/best/1$ izbere najboljšega posameznika iz trenutne populacije, dokler strategije $DE/best_z/1$ iz zgodovinske populacije. Način ustvarjanja mutiranega posameznika s pomočjo nove mutacije prikazujeta enačba (3.14) in algoritem 3.15.

$$\begin{aligned} \vec{m} &= \lfloor \vec{x}_{best}^{z,g} + F \cdot (\vec{x}_{r_1}^{t,g} - \vec{x}_{r_2}^{t,g}) + 0.5 \rfloor \\ r_1, r_2 &\in \{0, N_p - 1\}, \quad r_1 \neq r_2 \neq i \end{aligned} \quad (3.14)$$

Algoritem 3.15 Mutacija v algoritmu uglaševanja

Oznake: r_1, r_2 : naključno izbrani indeksi, i : indeks posameznika, j : indeks elementa vektorja, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, F : krmilni parameter mutacije, $\text{rand}\{0, N_p - 1\}$: funkcija, ki vrne naključno celo število na intervalu $[0, N_p - 1]$, m_j : j -ti element mutiranega vektorja, $x_{best}^{z,g}$: j -ti element najboljšega vektorja iz zgodovinske populacije. $x_{r_1,j}^{t,g}, x_{r_2,j}^{t,g}$: j -ti element vektorjev iz trenutne populacije na indeksih in r_1 in r_2 .

```

1:  $r_1 = \text{rand}\{0, N_p - 1\}$ ;
2: while  $r_1 == i$  do
3:    $r_1 = \text{rand}\{0, N_p - 1\}$ ;
4: end while
5:  $r_2 = \text{rand}\{0, N_p - 1\}$ ;
6: while  $r_2 == i$  or  $r_2 == r_1$  do
7:    $r_2 = \text{rand}\{1, N_p - 1\}$ ;
8: end while
9: for  $j = 0$  to  $D$  do
10:   $m_j = \lfloor x_{best,j}^{z,g} + F \cdot (x_{r_1,j}^{t,g} - x_{r_2,j}^{t,g}) + 0.5 \rfloor$ ;
11: end for

```

Da bi izboljšali nove posameznike, smo v mutaciji uporabili najboljšega posameznika iz zgodovinske populacije. Ta posameznik je odigral več iger proti različnim nasprotnikom in njegova ocena je bolj merodajna kot ocene posameznikov iz trenutne in poskusne populacije. Izbran pa je glede na njegovo učinkovitost oz. razmerje med doseženimi točkami in številom vseh odigranih iger, kot prikazuje enačba (3.15). Kot smo že omenili, posamezniki v algoritmu uglaševanja vsebujejo celoštevilčne elemente. Ker mutacija skalira vektor razlik za določeno realno vrednost, je potrebno elemente vektorjev še zaokrožiti na cela števila.

$$\text{učinkovitost}(\vec{x}_n^{z,g}) = \frac{r_n^{z,g}}{i_n^{z,g}}, \quad n = 1, 2, \dots \quad (3.15)$$

3.3.6 Zgodovinsko vbrizgavanje

Zgodovinsko vbrizgavanje je drugi del zgodovinskega mehanizma in je v algoritmu 3.10 prikazan v vrstici 14. Skozi evolucijski proces mehanizem zgodovinskega vbrizgavanja vrne potencialno dobre izumrle posameznike nazaj v evolucijski proces. Način delovanja tega mehanizma prikazuje algoritem 3.16. Najprej se nastavijo podatki o številu odigranih iger in doseženem rezultatu za vse na novo ustvarjene posameznike na 0. Nato pa se uporabi proporcionalna selekcija, ki izbere posameznike za vbrizgavanje. Proporcionalna selekcija za uspešnost posameznikov uporablja učinkovitost posameznikov (enačba 3.15). Posamezniki, ki imajo boljšo učinkovitost skozi evolucijski proces, imajo večjo verjetnost, da bodo izbrani in vbrizgani na določeno naključno mesto (m) v poskusno populacijo. Če pa je izbrani posameznik že v trenutni ali poskusni populaciji, ga ne vbrizgujemo. V tem mehanizmu smo vpeljali dva nova kontrolna parametra Z_{vel} in Z_{vbriz} . Prvi določa, koliko najboljših posameznikov bo uporabljenih v proporcionalni selekciji. Drugi parameter pa določa, koliko posameznikov bomo poskušali vbrizgati v poskusno populacijo.

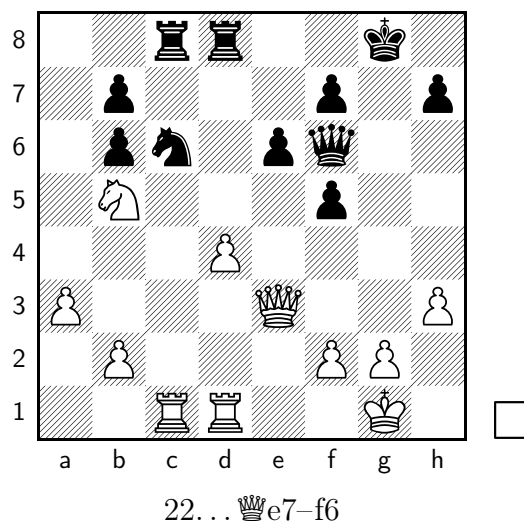
Algoritem 3.16 Zgodovinsko vbrizgavanje

Oznake: Z_g : zgodovinska populacija, g : številka trenutne generacije, Z_{vel} : število najboljših igralcev zgodovine, ki bodo uporabljeni v proporcionalni selekciji, Z_{vbriz} : število, ki določa število poskusov vbrizgavanja, $velikost$: število najboljših posameznikov, ki sodeluje pri proporcionalni selekciji, $x_l^{z,g}$: l -ti posameznik zgodovinske populacije, T_g : trenutna populacija, P_g : poskusna populacija, m : indeks poskusne populacije, kamor bo posameznik vbrizgan, N_p : velikost trenutne in poskusne populacije, D : dimenzija problema, $r_i^{z,g}$: dosežen rezultat i -tega posameznika v zgodovinski populaciji, $i_i^{z,g}$: število iger i -tega posameznika v zgodovinski populaciji, $r_i^{p,g}$: dosežen rezultat i -tega posameznika v poskusni populaciji, $i_i^{p,g}$: število iger i -tega posameznika v trenutni populaciji, $\text{rand}\{0, N_p - 1\}$: naključna števila iz množice $\{0, N_p - 1\}$ in $\vec{x}_m^{p,g}$: m -ti posameznik iz poskusne populacije.

```

1: for  $i = 0$  to  $N_p$  do
2:    $r_i^{p,g} = i_i^{p,g} = 0$ ;
3: end for
4:  $velikost = Z_{vel}$ ;
5: if  $velikost > velikostPopulacije(Z_g)$  then
6:    $velikost = velikostPopulacije(Z_g)$ ;
7: end if
8: for  $k = 0$  to  $Z_{vbriz}$  do
9:    $l = \text{proporcionalnaSelekcija}(Z_g, velikost)$ ;
10:  if not  $\text{populacijiVsebujeta}(T_g, P_g, \vec{x}_l^{z,g}, N_p, D)$  then
11:     $m = \text{rand}\{0, N_p - 1\}$ ;
12:     $\vec{x}_m^{p,g} = \vec{x}_l^{z,g}$ ;
13:     $i_m^{p,g} = i_l^{z,g}$ ;
14:     $r_m^{p,g} = r_l^{z,g}$ ;
15:  end if
16: end for

```



Šahovska partija je lahko raj ali pekel. Toda v tem rajju je včasih hudo in v tem peklu je včasih prijetno. (Mihail Talj)

4 Poskusi

Predstavljen algoritem uglaševanja smo preizkusili s pomočjo uglaševanja ocenitvene funkcije šahovskega programa BBChess 1.2. Zato v tem poglavju najprej podajamo podroben opis šahovskega programa in njegove ocenitvene funkcije. Nato sledi še opis načrta izvedbe poskusov in dobljenih rezultatov uglaševanja.

4.1 Šahovski program

BBChess¹ je šahovski program, ki je prosto dostopen odprto-kodni program in razvit pod pogoji licence GPL². Implementiran je v programskem jeziku ANSI C, temelji na programski knjižnici GLib³ in je posebej prilagojen za 64-bitne procesorje. Program ima implementirano bitno predstavitev deske (ang. *Bitboard*), MVV/LVA (ang. *Most Valuable Victim / Least Valuable Attacker*) generator potez, UCI (ang. *Universal Chess*

¹BBChess je dostopen na spletnem naslovi: <http://labraj.uni-mb.si/~borko/chess>

²GPL je licenca za prosto dostopno programsko opremo

³GLib je knjižnica, ki vsebuje programska orodja za različne platforme

Interface) vmesnik, ocenitveno funkcijo in iskalni algoritem, ki temelji na algoritmu NegaMax. Iskalni algoritem vključuje naslednje koncepte: iterativno poglobljanje (ang. *Iterative Deepening*), aspiracijska okna, (ang. *Aspiration Window Search*), klestenje z ničelno potezo (ang. *Null Move Pruning*), klestenje blizu listov (ang. *Futility Pruning*), iskanje mirovanja (ang. *Quiescence Search*), transpozicijsko tabelo (ang. *Transposition Table*) in paralelizacijo algoritma s pomočjo koncepta YBWC (ang. *Young Brothers Wait Concept*). Več o omenjenih predstavitev in algoritmih najdete v [34, 35, 10, 16, 51]. V našem poskusu bomo uglaševali najpomembnejši del programa tj. ocenitveno funkcijo. Njen podrobnejši opis sledi v naslednjem podpoglavju.

4.1.1 Ocenitvena funkcija šahovskega programa

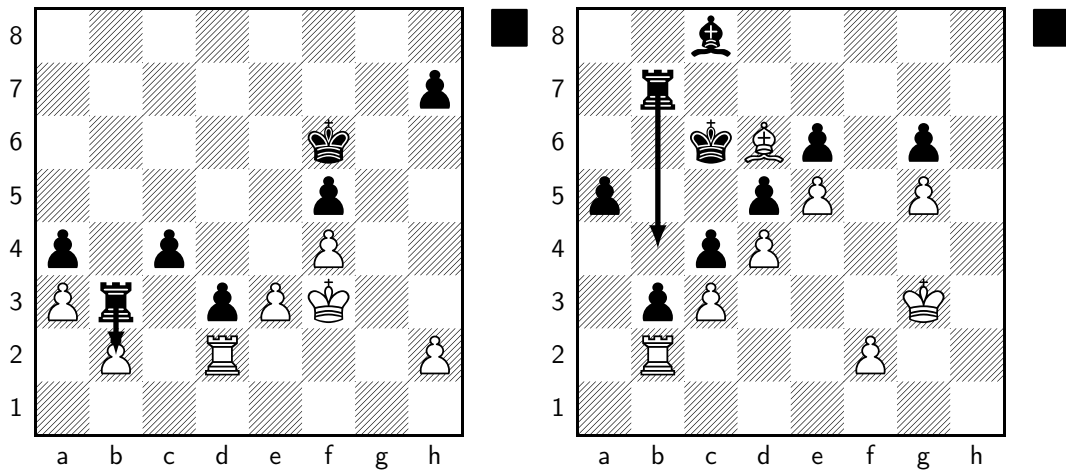
Ocenitvena funkcija podaja hevristične oz. statične ocene šahovskih pozicij. Na osnovi teh ocen iskalni algoritem preiskuje drevo igre in izbere najboljšo potezo za nadaljevanje. Zaradi velikosti drevesa igre iskalni algoritem v določenih vozliščih drevo odreže in oceni s pomočjo ocenitvene funkcije. Ocena vozlišča oz. pozicije je ponavadi celo število, ki določa, kateri od igralcev ima boljšo pozicijo oz. večje možnosti za zmago. Ocenitvena funkcija tako predstavlja jedro šahovskega programa in v veliki meri vpliva na obnašanje in igralno moč programa. Dobre ocenitvene funkcije vsebujejo določena šahovska znanja v obliki različnih aritmetičnih izrazov. Aritmetični izrazi so zasnovani tako, da ovrednotijo določene lastnosti v pozicijah. Obtežena vsota vseh lastnosti tako določa končno oceno pozicij. Opisan način delovanja ocenitvene funkcije prikazuje enačba (4.1) [19]. V enačbi oznaka pos predstavlja pozicijo, ki jo želimo oceniti, oznake f_1, f_2, \dots, f_n predstavljajo ocene različnih lastnosti pozicije in oznake w_1, w_2, \dots, w_n predstavljajo uteži pripadajoče lastnosti. Lastnosti pozicije so aritmetični izrazi, ki predstavljajo šahovsko znanje. V teh izrazih je mnogo parametrov, ki so nastavljeni intuitivno. To pomeni, da vrednosti parametrov niso nujno optimalne in da s pomočjo uglaševanja teh vrednosti lahko izboljšamo ocenitveno funkcijo in tudi igralno moč šahovskega programa.

$$eval(pos) = w_1 f_1(pos) + w_2 f_2(pos) + \dots + w_n f_n(pos) = \sum_{i=1}^n w_i f_i(pos) \quad (4.1)$$

Način delovanja ocenitvene funkcije ponazorimo na preprosti ocenitveni funkciji, ki vsebuje znanje o materialni prednosti in mobilnosti. Funkcijo prikazuje enačba (4.2).

$$eval(pos) = w_m(mob_{beli}(pos) - mob_{\text{\v{c}rni}}) + \sum_{i=0}^5 w_i(n_{i,beli}(pos) - n_{i,\text{\\v{c}rni}}(pos)) \quad (4.2)$$

V ocenitveni funkciji w_m je utež mobilnosti in njena vrednost je nastavljena na 10. Funkciji $mob_{beli}(pos)$ in $mob_{\text{\\v{c}rni}}(pos)$ izračunata mobilnost belih oz. \v{c}rnih figur. Oznake w_i predstavljajo uteži materialnih vrednosti figur in imajo naslednje vrednosti: 900 za damo, 500 za trdnjavo, 300 za lovca in skakača ter 100 za kmeta. Funkciji $n_{i,beli}(pos)$ in $n_{i,\text{\\v{c}rni}}(pos)$ pa izračunata število določenih figur v poziciji. Če s pomočjo predstavljene funkcije ocenimo poziciji prikazani na sliki 4.1.1, dobimo oceni, kot jih prikazujeta enačbi (4.3).



Slika 4.1: Primer ocenjevanja pozicij.

Obe oceni kažeta, da ima \v{c}rni nekoliko večje možnosti za zmago. Dejansko pa sta obe poziciji dobljeni za \v{c}rnega. Potezi, ki peljeta v zmago, sta b3b2 in b7b4. Iz tega vidimo, da statična ocena ni popolna in da za končno oceno pozicije potrebujemo iskalni algoritem. Zaradi učinka obzorja pa lahko tudi iskalni algoritem napačno oceni pozicijo.

$$eval(pos_l) = 10(12 - 16) + 100(5 - 5) + 500(1 - 1) = -40 \quad (4.3)$$

$$eval(pos_d) = 10(22 - 13) + 100(5 - 6) + 300(1 - 1) + 500(1 - 1) = -10$$

Na enak način je zasnovana tudi ocenitvena funkcija, ki jo bomo uglaševali. Ta ocenitvena funkcija je kompleksnejša in odkriva naslednje lastnosti v pozicijah:

- material figur,
Predstavlja razliko med vrednostjo belih in črnih figur. Npr. bolje je imeti damo kot trdnjavo.
- položaj figur,
Vrednosti položajev figur. Npr., če figure zasedajo sredino deske je njihova ocena boljša.
- zaščito kralja:
 - napad na kralja,
Napad nasprotnikovih figur na kralja. Npr. oddaljenost nasprotnikove dame od kralja.
 - zaklonišče,
Kako je kralj zaščiten s pomočjo svojih kmetov? Npr., če je kralj na polju g1, na poljih f2, g2 in h2 pa ni kmetov, potem je kralj slabo zaščiten.
 - nevihta kmetov,
Bližina nasprotnikovih kmetov, kralju. Npr., če je kralj na polju g1, na poljih f2, g2, h2, f3, g3 in h3 pa je več kot eden nasprotnikovih kmetov, potem je zaščita kralja slabša.
 - možnost rokade,
Kam imamo možnost opraviti rokado in kakšna bi bila zaščita kralja po rokadi.
- vzorci figur,
Vzorci figur, ki jih težko ocenimo s pomočjo ocenitvene funkcije in iskalnim algoritmom. Npr. ujet lovec na drugi oz. sedmi liniji.
- formacija kmetov:
 - zaostali kmeti,
Kmet je zaostal za ostalimi kmeti in tako lahko postane lahek plen nasprotnikovih figur.
 - kandidati za promocijo,
Kmet ima prosto pot do osme linije in lahko pride do ti. promocije.

- izolirani kmeti,

Kmet nima sosednjih kmetov in lahko postane lahek plen nasprotnikovih figur.

- verjetnosti za zmago v določenih končnicah.

Izračunajo se verjetnosti za zmago v določenih končnicah. Npr., če vsak od igralcev ima lovca, kralja in kmete je pozicija verjetno remi, kadar sta lovca na poljih različnih barv in materialna razlika ni večja od dveh kmetov.

4.2 Načrt poskusov

Da bi ugotovili učinkovitost našega algoritma, smo ga preizkusili s pomočjo uglasitve naše ocenitvene funkcije na različnih velikostih intervalov. Najprej smo algoritem preizkusili pri uglasitvi materialnih vrednosti figur in mobilnosti na ekstremno širokih intervalih. Cilj tega poskusa je pokazati ali je algoritem sposoben uglasiti parametre brez uporabe ekspertnega znanja oz. uglasiti parametre, za katere niti približno ne vemo, kje je njihov optimum.

Razvijalci šahovskih programov ponavadi poznajo približne vrednosti parametrov in tako lahko določijo intervale, na katerih se optimalne vrednosti nahajajo. Zato smo algoritem preizkusili s pomočjo uglasitve večjega števila parametrov na ozkih in nekoliko širših intervalih. Pri uglasitvi, kjer smo uporabili ozke intervale, smo preizkusili še vpliv števila odigranih iger na uglasitve. S pomočjo teh intervalov smo naš algoritem primerjali še s podobnim evolucionim algoritmom [30, 29, 24]. Na koncu smo algoritem preizkusili še na nekoliko širših intervalih uglasitve. Tako smo pokazali, kako se algoritem obnaša, ko so vrednosti parametrov bolj oddaljene od optima in ali bo izboljšava tako uglasenega programa večja kot v primeru ožjih intervalov.

Vse poskuse smo izvedli na računalniku s procesorjem Intel Core 2 CPU 2,4 GHz (dve jedri), 1 GB delovnega pomnilnika in operacijskim sistemom Mandriva 2008.0 x86_64. Program je bil preveden s pomočjo prevajalnika GCC 4.2.2. Pri prevajanju so bile uporabljene optimizacije O3 in optimizacije za Intel x86_64 arhitekturo. Vse poskuse smo izvajali paralelno oz. uporabljali smo obe jedri procesorja. Poskusi so trajali približno 6 mesecev.

4.3 Uглаševanje brez ekspertnega znanja

Ali je algoritem sposoben uglašiti parametre brez uporabe ekspertnega znanja oz. uglašiti parametre, za katere niti približno ne vemo, kje so njihove optimalne vrednosti? Da bi odgovorili na to vprašanje, smo algoritem preizkusili s pomočjo uglaševanja materialnih vrednosti figur in mobilnosti na ekstremno širokih intervalih. Materialne vrednosti figur so vrednosti figur kmeta, skakača, lovca, trdnjave in dame v otvoritveni in končni fazi igre (5 parametrov, ki se uporabljajo v otvoritveni fazi igre in 5 parametrov, ki se uporabljajo v končni fazi igre). Mobilnost pa določa, kako je določena figura mobilna. To je število, ki predstavlja število možnih potez določene figure. V našem primeru smo uporabili mobilnost skakača, lovca in trdnjave v otvoritveni in končni fazi igre (6 parametrov).

Za vse parametre smo določili intervale uglaševanja $[0, 1000]$. Definirani intervali omogočajo izbiro vrednosti parametrov tako, da ocenitvena funkcija ocenjuje pozicije popolnoma napačno. Napaka je lahko večja tudi od vrednosti štirih kmetov. Zato smo algoritmu odstranili pogoj za končanje igre, kjer ima eden od igralcev prednost 4 kmetov (glej podpoglavje 3.3.2). Prav tako uglaševalni program oz. posamezniki niso uporabljali otvoritvene knjižnice in baz končnic. Brez uporabe otvoritvene knjižnice in baz končnic se parametri uglašujejo v vseh fazah igre. To omogoča v tem poskusu hitrejšo konvergenco in učinkovitejše uglaševanje.

Tabela 4.1: Nastavitve eksperimenta pri uglaševanju materialnih vrednosti figur in mobilnosti

strategija	DE/best _h /1	Z_{vel}	20
N_p	40	Z_{vbriz}	4
D	16	globina	4
F	0,9	niti	2
C_r	0,9	predpomnilnik	16 MB
G	500	J_r	0,1
igre	5		

Nastavitve algoritma so prikazane v tabeli 4.1. Poskus smo zagnali skozi 500 generacij, kjer sta se dve igri igrali paralelno. Globina iskanja posameznikov je bila 4, velikost

predpomnilnika, ki so ga uporabljali posamezniki, pa je bila 16 MB. Uporabljena je bila strategija DE/best_n/1, velikost populacije je bila 40 in število parametrov oz. dimenzija problema je bila 16. Tako nastavljeni algoritem hitro konvergira in lahko zaide v lokalni optimum. Z večjo populacijo bi algoritem bil bolj robusten, toda znatno bi se povečala njegova časovna zahtevnost. Zato smo parametra mutacije in križanja nastavili na 0,9 in tako povečali robustnost algoritma. Krmilni parametri Z_{vel} , Z_{vbriz} in J_r so bili nastavljeni na vrednosti 20, 4 in 0,1. Število iger pri ocenjevanju posameznikov je bilo nastavljeno na 5. Pri inicializaciji so bili parametri nastavljeni uniformno naključno na intervalu $[0 - 1000]$. Povprečne vrednosti parametrov trenutne populacije po inicializaciji prikazuje tabela 4.2, vrednosti standardne deviacije pa tabela 4.3. Parametre smo označili z imeni in oznakami v oklepajih. Oznaki (o) in (k) predstavljata parametre materialne vrednosti figur v otvoritveni oz. končni fazi igre. Oznaki (mo) in (mk) pa predstavljata parametre mobilnost v otvoritveni oz. končni fazi igre.

Tabela 4.2: Povprečne vrednosti parametrov pri inicializaciji

kmet (o) = 457,88	trdnjava (k) = 532,27
skakač (o) = 511,05	dama (k) = 477,62
lovec (o) = 532,15	skakač (mo) = 453,65
trdnjava (o) = 413,30	skakač (mk) = 501,20
dama (o) = 533,25	lovec (mo) = 446,20
kmet (k) = 531,83	lovec (mk) = 505,65
skakač (k) = 527,58	trdnjava (mo) = 468,62
lovec (k) = 538,83	trdnjava (mk) = 513,95

Obnašanje algoritma skozi evolucijski proces prikazujeta sliki 4.2 in 4.3. Slika 4.2 prikazuje povprečne vrednosti parametrov. Iz grafa je razvidno, da je algoritem že skozi prvih 100 generacij ugotovil približne materialne vrednosti figur z izjemo parametra trdnjave v končnici igre in da imajo parametri mobilnosti male vrednosti. V prvih nekaj generacijah je algoritem najprej uglasil mobilnosti in šele nato začel uglaševati materialne vrednosti figur. Pri tem je algoritem zašel v lokalni optimum. Parameter trdnjave v končnici igre je imel napačno vrednost glede na ostale vrednosti parametrov.

Tabela 4.3: Standardne deviacije parametrov pri inicializaciji

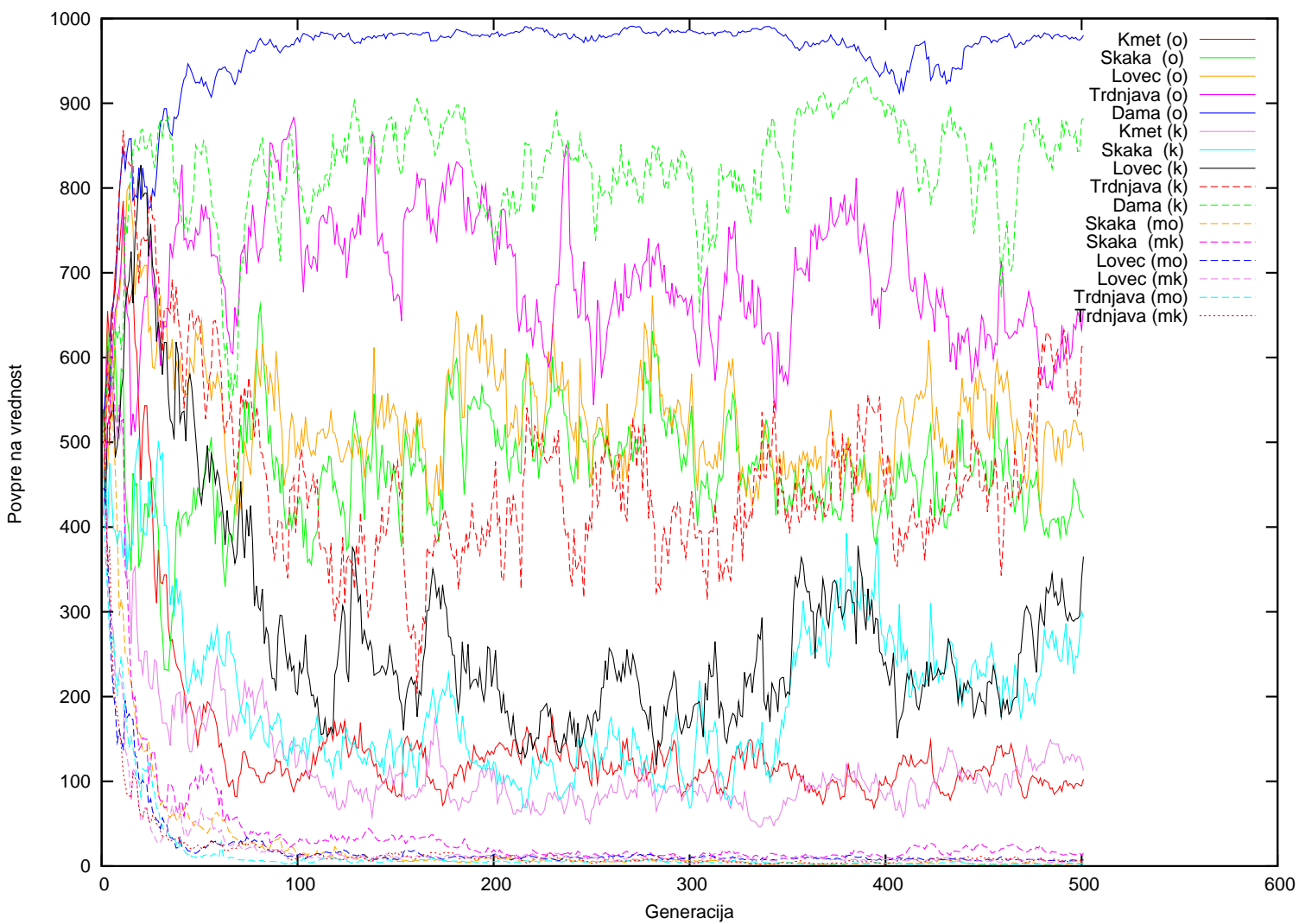
kmet (o) = 1915,14	trdnjava (k) = 1777,22
skakač (o) = 1902,34	dama (k) = 1804,27
lovec (o) = 1672,70	skakač (mo) = 1908,02
trdnjava (o) = 1537,03	skakač (mk) = 1949,91
dama (o) = 1631,07	lovec (mo) = 2026,91
kmet (k) = 1770,59	lovec (mk) = 1894,11
skakač (k) = 1923,22	trdnjava (mo) = 1815,12
lovec (k) = 1918,45	trdnjava (mk) = 1971,92

Njegova vrednost je bila manjša od vrednosti parametrov skakača in lovca v otvoritveni fazi igre. Iz slike 4.3 vidimo, da so standardne deviacije mobilnosti že pri stoti generaciji imele nižje vrednosti. Na drugi strani se vrednosti standardne deviacije materialnih vrednosti figur niso tako zmanjšale in variirajo. Razlog temu je napačna vrednost parametra trdnjave v končnici igre.

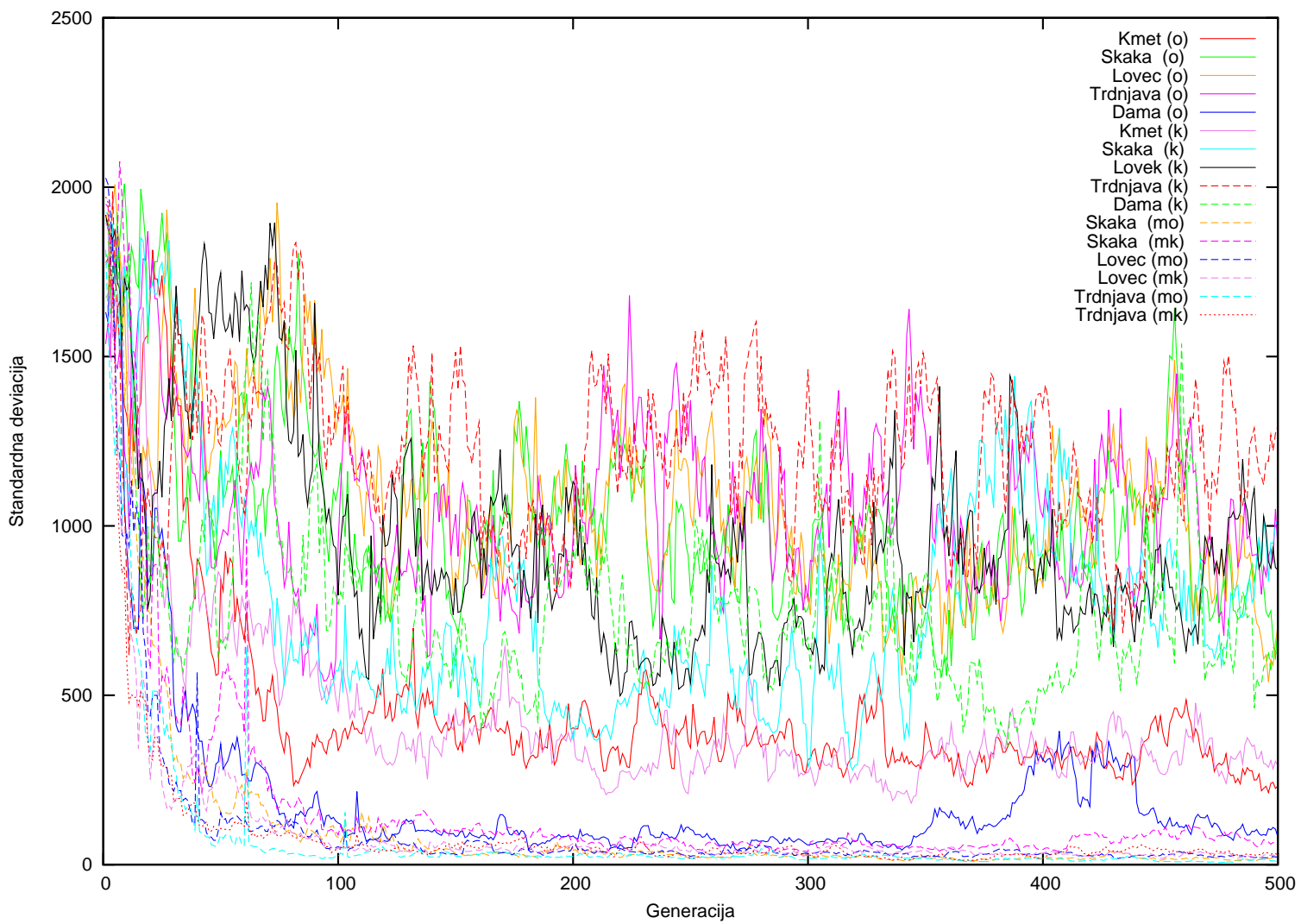
Tabela 4.4: Povprečne vrednosti zadnje generacije

kmet (o) = 102,53	trdnjava (k) = 614,35
skakač (o) = 410,80	dama (k) = 882,00
lovec (o) = 489,43	skakač (mo) = 3,17
trdnjava (o) = 657,55	skakač (mk) = 15,45
dama (o) = 980,00	lovec (mo) = 7,12
kmet (k) = 112,50	lovec (mk) = 5,78
skakač (k) = 292,93	trdnjava (mo) = 2,25
lovec (k) = 365,12	trdnjava (mk) = 8,28

Skozi evolucijski proces algoritmu uspe uglasiti parametre tako, da ustrezajo znanemu razmerju iz teorije šaha. To je razvidno iz povprečnih vrednosti parametrov trenutne populacije v končni generaciji, ki jih prikazuje tabela 4.4. Vrednosti standardne deviacije parametrov končne generacije prikazuje tabela 4.5. Vidimo, da ima parameter



Slika 4.2: Povprečne vrednosti parametrov



Slika 4.3: Standardne deviacije parametrov

Tabela 4.5: Standardne deviacije zadnje generacije

kmet (o) = 237,11	trdnjava (k) = 1503,02
skakač (o) = 527,20	dama (k) = 558,05
lovec (o) = 769,44	skakač (mo) = 19,28
trdnjava (o) = 868,47	skakač (mk) = 72,57
dama (o) = 87,49	lovec (mo) = 26,05
kmet (k) = 347,58	lovec (mk) = 26,17
skakač (k) = 1100,99	trdnjava (mo) = 11,02
lovec (k) = 1085,76	trdnjava (mk) = 40,22

trdnjave v končnici igre največjo vrednost standardne deviacije. To je tudi parameter, s katerim je algoritem imel največ težav pri uglasovanju.

Glede na dobljene rezultate lahko ugotovimo, da je algoritem uspešno uglasil 16 parametrov na ekstremno širokih intervalih. Algoritem je zelo hitro ugotovil približna razmerja parametrov, z izjemo parametra trdnjave v končnici igre. S tem parametrom je algoritem imel tudi največ težav. Vseeno pa ga je na koncu uspešno uglasil.

4.4 Uglasovanje na ozkih intervalih

V tem podpoglavju predstavljamo poskus uglasovanja večjega števila parametrov na ozkih intervalih. Intervale smo nastavili intuitivno, kot prikazuje tabela 4.22. S pomočjo tega poskusa smo pokazali obnašanje algoritma, kadar poznamo približne vrednosti parametrov oz. vrednosti blizu optimuma. Največji problem, ki ga mora algoritem v tem primeru rešiti, je ugotoviti, kateri posamezniki so boljši in kateri slabši. Razlog temu so vrednosti parametrov posameznikov, ki so nastavljeni blizu optimuma. To pomeni, da so razlike med parametri posameznikov manjše, prav tako tudi razlike v njihovi igralni moči. Zato je na osnovi nekaj iger zelo težko ugotoviti, kateri posamezniki so boljši in kateri slabši. Tak problem imajo ponavadi tudi razvijalci šahovskih programov. Parametre nastavijo intuitivno blizu optimalnih vrednosti, nato pa jih z uglasovanjem poskušajo izboljšati.

Algoritem je bil preizkušen s pomočjo štirih različnih nastavitvev. Za vsako nastavitvev smo zagnali 10 neodvisnih zagonov. Da bi pokazali učinkovitost algoritma, je bila v treh nastavitvah uporabljena različna vrednost krmilnega parametra *igre*. V četrti nastavitvi smo algoritem preizkusili brez uporabe zgodovinskega mehanizma. Tako smo pokazali vpliv zgodovinskega mehanizma na proces uglaševanja. Pri zagonih, kjer je bil uporabljen zgodovinski mehanizem, so bile uporabljene nastavitve, prikazane v tabeli 4.6. Pri zagonih brez zgodovinskega mehanizma so bile uporabljene nastavitve prikazane v tabeli 4.12. Pri vseh zagonih smo v procesu inicializaciji uporabili psevdonaključni generator števil, implementiran v standardnih Linux C knjižnicah z različnim semenom. Zagoni so bili omejeni na 100 generacij, uporabljena je bila otvoritvena knjižnica *performance.bin*, katere avtor je Marck Lacrosse in baze končnic petih figur *Scorpio bitbases*, katerih avtor je Daniel Shawul.

Otvoritvena knjižnica in baze končnic omogočajo programu, da v otvoritveni fazi igre in končnici igra brezhibno. Otvoritvena knjižnica omogoča izbiro različnih otvoritvenih variant, ki se uporabljajo v procesu uglaševanja. Za določeno pozicijo šahovski program vedno vrne isto najboljšo potezo. Tako bi se brez uporabe otvoritvene knjižnice program uglaševal le na nekaj otvoritvenih variant. Otvoritvena knjižnica programu za otvoritvene pozicije ponuja poteze in njihove verjetnosti. Tako program glede na ponujene verjetnosti izbere eno od potez. Uporabljena otvoritvena knjižnica je bila ustvarjena s pomočjo orodja PolyGlot. To orodje je prosto dostopno in v osnovi se uporablja kot UCI adapter. Omogoča povezovanje UCI šahovskih programov z grafičnimi vmesniki, ki podpirajo XBoard protokol. Dodatno pa to orodje lahko uporabljamo za ustvarjanje otvoritvenih knjižnic. Verjetnosti za izbiro poteze v otvoritveni knjižnici orodje določa glede na pogostost izbire poteze v podanih igrah in glede na izide iger. Otvoritvena knjižnica lahko vsebuje znanja oz. izkušnje glede na igre, ki so jih odigrali ljudje ali programi. Uporabljena knjižnica je bila zgrajena na osnovi iger odigranih med ljudmi in šahovskimi programi. Število iger odigranih med ljudmi je bilo približno enako, kot število iger odigranih med programi [40].

V končni fazi igre, ko pozicija vsebuje 5 ali manj figur, program uporablja baze končnic namesto ocenitvene funkcije. To programu omogoča, da v takih pozicijah igra brezhibno. Zaradi tega se v procesu uglaševanja igra konča v poziciji končnice. Rezultat

igre določa ocena shranjena v bazi končnice.

Uporaba otvoritvene knjižnice in baz končnic zmanjša časovno zahtevnost algoritma uglaševanja. V otvoritveni fazi igre se izbrana poteza vrne hitreje kot, če uporabimo iskalni algoritem. Baze končnic omogočajo, da se igre hitreje zaključijo. Tako se v procesu uglaševanja večinoma uglašujejo parametri, ki se uporabljajo za ocenjevanje pozicij v srednji fazi igre. Z uporabo teh dveh komponent smo ustvariti tudi enake pogoje, kot v igrah, ko program igra proti različnim nasprotnikom zunaj evlucijskega procesa.

V tem poskusu so bile vrednosti parametrom trenutne populacije pri inicializaciji nastavljene uniformno-naključno na definiranih intervalih (tabela 4.22, str. 70). Vrednosti parametrom poskusne populacije so bile izračunane s pomočjo mehanizma nasprotij (glej podpoglavje 3.2.1). Uporabljena je bila mala velikost populacije ($N_p = 20$). Razlog temu je časovna zahtevnost algoritma in potreba po hitri konvergenci. Pri ocenjevanju posameznikov sta se uporabljali dve niti oz. dve igri sta se igrali istočasno. Pri igranju iger so posamezniki uporabljali preiskovanje z globino 6. Manjše globine iskanja ne bi bistveno izboljšale časovno zahtevnost algoritma. Razlog temu je iskalni algoritem, ki uporablja metodo klestenja vozlišč, ki so blizu listov preiskovalnega drevesa (ang. *Futility Pruning*) [35]. Manjše globine preiskovanje bi povzročile tudi večji šum v procesu uglaševanja. Slabši posamezniki bi imeli večjo verjetnost, da bi premagali boljše posameznike. Na drugi strani bi večje globine preiskovanja zaradi narave iskalnega algoritma znatno povečale časovno zahtevnost algoritma uglaševanja. Posamezniki so uporabljali 16 MB predpomnilnika in enako velikost predpomnilnika namenjenega podatkovnim bazam končnic.

Zaradi časovne zahtevnosti algoritma je težko ugotoviti optimalne vrednosti krmilnih parametrov. Zato smo njihove vrednosti nastavili glede na priporočila, predstavljena v nadaljevanju tega odstavka. Krmilna parametra F in C_r sta bila nastavljena na vrednosti 0,5 in 0,9. Ti vrednosti smo nastavili glede na priporočila iz literature [57], kjer je predlagano, da sta ti vrednosti dobri za globalno optimizacijo. Krmilni parameter J_r je bil nastavljen na vrednost 0,1 glede na priporočila iz literature [50]. Za

neznan optimizacijski problem so za J_r predlagali vrednost, ki se nahajajo na intervalu $[0, 1 - 0, 4]$. Kontrolna parametra Z_{vel} in Z_{vbriz} sta nastavljena na vrednosti 20 in 4. Ti vrednosti smo nastavili z uporabo enačb $Z_{vel} = N_p$ in $Z_{vbriz} = \frac{N_p}{5}$. Zaustavitveni pogoj evolucijskega procesa v algoritmu predstavlja število generacij, ki je bilo nastavljeno na 100.

4.4.1 Vpliv šuma

Da bi ugotovili, kako vpliva šum na algoritem, smo ga preizkusili s pomočjo treh različnih nastavitvev parametra *igre*. Ta parameter določa število iger, ki jih posamezniki odigrajo v evolucijskem procesu. Več iger kot jih odigrajo, večja je verjetnost, da njihove ocene vsebujejo manjšo napako oz. šum. Vrednosti parametra *igre* so bile nastavljene na 2, 5 in 10. Vsaka od teh vrednosti je bila uporabljena v deset neodvisnih zagonih. Ostale nastavitve algoritma prikazuje tabela 4.6.

Tabela 4.6: Nastavitve parametrov pri algoritmu, ki uporablja zgodovinski mehanizem

strategija	DE/best _n /1	Z_{vel}	20
N_p	20	Z_{vbriz}	4
D	190	<i>globina</i>	6
F	0,5	niti	2
C_r	0,9	predpomnilnik	16 MB
G	100	otvoritvena knjižnica	performance.bin
J_r	0,1	baze končnic	Scorpio bitbases
<i>igre</i>	10, 5, 2	predpomnilnik baz končnic	16 MB

Po končanih zagonih je bilo potrebno ugotoviti rezultat uglasovanja oz. za koliko smo izboljšali kvaliteto ocenitvene funkcije. To smo naredili tako, da smo vzeli najbolje ocenjenega posameznika iz začetne populacije (Začetni) in 20 najbolj ocenjenih posameznikov (Uglašen) iz zgodovinske populacije v končni generaciji. Najboljši posameznik iz začetne populacije ima najboljšo učinkovitost (enačba 3.15) v zgodovinski populaciji pri inicializaciji. Po končanem evolucijskem procesu ne vemo, kateri posameznik je najboljši. Določeni posamezniki so preživeli le nekaj generacij in tako odigrali malo število iger. Za te posameznike ne vemo, ali so dejansko dobri. Na osnovi malega

števila iger je praktično nemogoče določiti igralno moč posameznika. To je razlog, da smo po končanem evolucionem procesu vzeli 20 najboljših posameznikov po učinkovitosti (enačba 3.15) iz zgodovinske populacije. Izbrane posameznike smo nato ocenili s pomočjo programa Rybka 2.3.2a.

Ocenjevanje je potekalo tako, da so posamezniki odigrali dvoboj s 50 igrami proti programu Rybka, katerega igralna moč je bila omejena na 1700, 1900, 2100 in 2300 rating točk. Pri igranju iger je bila uporabljena globina preiskovanja 6 in vsak posameznik je v dvoboju odigral 25 iger z belimi figurami in 25 iger s črnimi figurami. Na osnovi odigranih iger smo nato ocenili igralno moč posameznika s pomočjo algoritma Minorization-Maximization [37], ki je implementiran v prosto dostopnem orodju Bayeselo. Za izračun rezultatov enega zagona je bilo odigranih 4200 iger. En zagon je trajal približno 10 ur. Tako je ocenjevanje 10 zagonov trajalo približno 4 dni oz. za vse tri nastavitve algoritma 12 dni. Dobljene rezultate prikazujejo tabele 5.1, 5.2 in 5.3 (priloga str.: 82, 85 in 88), v krajši obliki pa tabele 4.7, 4.8 in 4.9.

Tabela 4.7: Rezultati uglaševanja ($igre = 2$)

	1. zagon	2. zagon	3. zagon	4. zagon	5. zagon	6. zagon	7. zagon	8. zagon	9. zagon	10. zagon
Rybka ₂₃₀₀	2300±25	2300±22	2300±24	2300±24	2300±23	2300±22	2300±23	2300±24	2300±26	2300±25
Uglašen ₁ ²	2073±48	2130±49	2085±48	2066±49	2079±49	2139±48	2101±48	2089±48	2071±48	2040±48
Rybka ₂₁₀₀	2128±20	2199±20	2154±20	2168±20	2175±20	2161±19	2174±20	2113±20	2178±22	2087±19
Rybka ₁₉₀₀	1957±19	1953±19	1982±19	1938±19	1923±19	1985±19	1971±19	1943±19	2018±19	1918±19
Začetni ²	1925±49	1854±53	1909±50	1862±53	1914±50	1983±47	1933±49	1853±51	1980±49	1891±50
Rybka ₁₇₀₀	1786±21	1751±24	1741±22	1713±24	1723±24	1735±24	1762±23	1705±23	1740±21	1681±24
Izboljšava	148	276	176	204	165	156	168	236	91	149

Tabela 4.8: Rezultati uglaševanja ($igre = 5$)

	1. zagon	2. zagon	3. zagon	4. zagon	5. zagon	6. zagon	7. zagon	8. zagon	9. zagon	10. zagon
Rybka ₂₃₀₀	2300±22	2300±22	2300±22	2300±21	2300±22	2300±23	2300±23	2300±22	2300±22	2300±23
Uglašen ₁ ⁵	2127±50	2131±47	2116±48	2177±48	2116±47	2109±48	2170±48	2147±49	2111±46	2124±47
Rybka ₂₁₀₀	2135±19	2174±20	2138±19	2161±19	2170±20	2117±19	2134±19	2135±19	2146±19	2150±20
Rybka ₁₉₀₀	1930±19	1979±19	1946±19	1970±20	1984±19	1925±19	1952±19	1952±19	1965±19	2007±19
Začetni ⁵	1907±48	1966±48	1872±51	1745±58	1963±50	1755±54	1782±54	1961±49	2044±47	1874±50
Rybka ₁₇₀₀	1727±25	1807±22	1750±24	1819±23	1746±24	1714±24	1741±24	1727±25	1772±23	1771±22
Izboljšava	220	165	244	432	153	354	388	186	67	250

Tabela 4.9: Rezultati uglaševanja (*igre* = 10)

	1. zagon	2. zagon	3. zagon	4. zagon	5. zagon	6. zagon	7. zagon	8. zagon	9. zagon	10. zagon
Rybka ₂₃₀₀	2300±21	2300±21	2300±21	2300±23	2300±23	2300±24	2300±22	2300±21	2300±22	2300±22
Uglašen ₁ ¹⁰	2145±48	2151±48	2195±48	2114±49	2132±48	2082±48	2202±48	2152±49	2125±48	2127±46
Rybka ₂₁₀₀	2157±19	2163±20	2191±19	2159±20	2166±20	2128±20	2147±19	2189±19	2184±20	2146±19
Rybka ₁₉₀₀	1957±19	1949±19	2017±19	1985±19	1959±19	1930±19	1983±19	1977±19	1957±19	1997±19
Začetni ¹⁰	1902±49	1940±50	1975±48	1905±54	1760±57	1860±50	1926±48	1974±49	1979±50	1927±50
Rybka ₁₇₀₀	1733±23	1781±23	1800±23	1666±26	1753±23	1691±24	1787±23	1701±27	1763±23	1746±24
Izboljšava	243	211	220	209	372	222	276	178	146	200

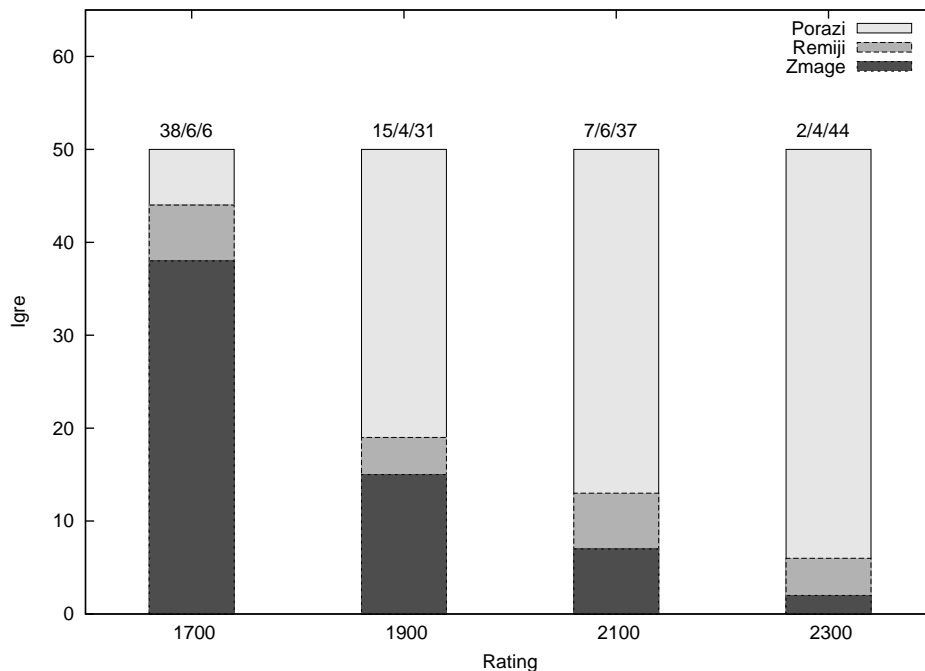
V tabelah so najboljše ocenjeni uglašeni posamezniki označeni z Uglašen₁², Uglašen₁⁵ in Uglašen₁¹⁰. Najbolje ocenjeni posamezniki začetne populacije pa so označeni z Začetni², Začetni⁵ in Začetni¹⁰. Nad-pisana številka določa uporabljeno vrednost krmilnega parametra *igre* v procesu uglaševanja. Oznake Rybka₁₇₀₀, Rybka₁₉₀₀, Rybka₂₁₀₀ in Rybka₂₃₀₀ predstavljajo program Rybka z omejeno močjo 1700, 1900, 2100 in 2300 rating točkami.

Vrednosti v tabelah predstavljajo rating točke in 95% intervale zaupanja (ang. *Confidence Intervals*), ki smo jih izračunali s pomočjo orodja Bayeselo. Zaradi lažje primerjave smo vse rating točke premaknili tako, da je Rybka₂₃₀₀ imela 2300 rating točk. Zadnja vrstica v tabelah prikazuje izboljšavo, v obliki rating točk, najboljše ocenjenega posameznika uglaševanja glede na najboljše ocenjenega posameznika iz začetne populacije. Iz rezultatov vidimo, da je bilo uglaševanje uspešno v vseh zagonih in da so dosežene izboljšave bile od 86 do 432 rating točk.

Tabela 4.10: Vpliv krmilnega parametra *igre*

	Najboljši rating	Povprečni rating (σ)	Najboljša izboljšava	Povprečna izboljšava (σ)	Povprečni začetni rating (σ)	Število iger
Uglašen ₁ ¹⁰	2202±48	2142,5 (35,89)	372	227,7 (61,48)	1914,8 (66,15)	40000
Uglašen ₁ ⁵	2177±48	2132,8 (24,17)	432	245,9 (114,49)	1886,9 (100,86)	20000
Uglašen ₁ ²	2139±48	2087,3 (29,67)	276	176,9 (51,31)	1910,4 (47,2)	8000

Tabela 4.10 prikazuje vpliv krmilnega parametra *igre* na proces uglaševanja. V tabeli prikazujemo najboljše in povprečno dosežene ratinge ter najboljše in povprečne izboljšave. Vidimo, da večje vrednosti parametra *igre* omogočajo doseganje boljših ratingov.



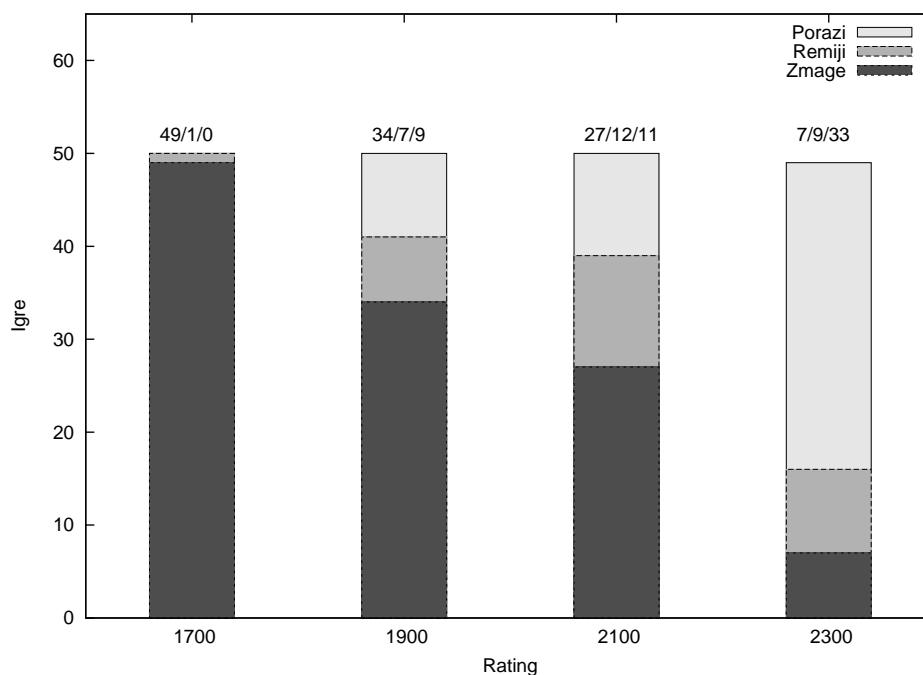
Slika 4.4: Histogram zmag, remijev in porazov najbolje ocenjenega posameznika iz začetne populacije pri 7. zagonu.

Razlog temu je šum, ki se zmanjšuje z večjim številom odigranih iger in omogoča boljše ocenjevanje posameznikov. Najboljše izboljšave ne kažejo na korelacijo s krmilnim parametrom *igre*. Najboljše izboljšave so dosežene, ko je krmilni parameter *igre* imel vrednosti 5. Razlog temu so najbolje ocenjeni posamezniki iz začetne populacije, ki so dosegli manjše ratinge. Največji vpliv so imeli 4., 5. in 6. zagon. Tukaj so bile dosežene največje izboljšave (432, 354 in 388 rating točk) in posamezniki iz začetne populacije so dobili najslabše ocene (1745, 1755 in 1782 rating točk).

4.4.2 Izboljšava in moč najbolje ocenjenega posameznika

V tem podpoglavju podrobneje predstavimo izboljšavo uglaševanja in doseženo igralno moč posameznika z največjim doseženim ratingom glede na igre, ki so jih posamezniki igrali proti programu Rybka (Tabele 4.7, 4.8, 4.9) in glede na odigran turnir proti odprto-kodnima programoma Crafty in GNU Chess.

Najboljši rating smo dosegli v sedmem zagonu, ko je bil krmilni parameter *igre* nastavljen na 10. Rating tega posameznika je znašal 2202 ± 48 rating točk, njegovi parametri



Slika 4.5: Histogram zmag, remijev in porazov najbolje ocenjenega posameznika iz 7. zagona.

pa so prikazani v tabeli 4.22. Izboljšava glede na najbolje ocenjenega posameznika iz začetne populacije je znašala 276 rating točk. Pri ocenjevanju sta posameznika igrala igre proti programu Rybka z omejeno močjo. Rezultate teh iger prikazujeta histograma na slikah 4.4 in 4.5. Kot vidimo je bil, najbolje ocenjen posameznik uspešnejši proti vsem nasprotnikom. Oba posameznika sta dosegla dober rezultat proti programu Rybka z močjo 1700 rating točk. Posameznik iz začetne populacije je imel 36 zmag, 6 remijev in 6 porazov. Najbolje uglašeni posameznik je zmagal vse igre, razen ene, ki pa se je končala z remijem. Proti programu Rybka z močjo 1900 rating točk je posameznik iz začetne populacije dosegel bistveno slabši rezultat glede na najbolje ocenjenega posameznika uglaševanja. Zmagal je samo 15 iger, najbolje ocenjeni uglašeni posameznik pa 34 iger. Proti programu Rybka z igralno močjo 2100 in 2300 rating točk je posameznik iz začetne populacije izgubil skoraj vse igre. Najbolje uglašeni posameznik je dosegel pozitiven rezultat proti programu Rybka z močjo 2100 rating točk tj. 27 zmag, 11 remijev in 12 porazov. Negativni rezultat je dosegel le proti programu Rybka z močjo 2300 rating ročk, kjer je zmagal 7 iger, remiziral 9 iger in poražen je bil v 33-ih igerah.

Najbolje ocenjeni uglašeni posameznik in najbolj ocenjeni posameznik iz začetne populacije sta igrala turnir proti znanima odprto-kodnima programoma Crafty-21.6 in GNU Chess 5.07. Na turnirju je vsak igralec igral proti vsakemu igralcu 100 iger s časovno omejitvijo 5 minut za 40 potez. BBChess je uporabljal otvoritveno knjižnico *performace.bin* in baze končnic (*Scorpio bitbases*) 5 figur. Uporabljal je še paralelni iskalni algoritem in tako izkoriščal obe jedri procesorja. Program Crafty je uporabljal svojo otvoritveno knjižnico *book.bin* in svoje baze končnic (*tablebases*) 5 figur. Programa GNU Chess in Crafty sta preiskovala v *pondering* načinu tj. “razmišljala” sta tudi, ko nista bila na potezi. Dosežene rezultate turnirja prikazuje tabela 4.11 in izračunani so bili s pomočjo orodja Bayeselo.

Tabela 4.11: Turnir – 5 minut za 40 potez.

Mesto	Ime	Rating	\pm	Igre	Rezultat	Nasprotnik	Remi
1	Crafty-21.6	2647	45	300	86%	2329	15%
2	Uglašen ¹⁰	2458	37	300	58%	2391	19%
3	Začetni ¹⁰	2309	39	300	35%	2441	15%
4	GNU Chess 5.07	2219	42	300	22%	2471	13%

V tabeli 4.11 so prikazani doseženi ratingi, 95% intervali zaupanja, število odigranih iger, dosežene uspešnosti, povprečni ratingi nasprotnikov in odstotki remijev. Ratingi so bili izračunani glede na rating programa Crafty, ki je bil nastavljen na 2646 točk glede na CCRL (Computer Chess Rating List) lestvico. To lestvico oblikuje klub zanesenjakov, ki radi opazujejo igranje šahovskih računalniških programov, jih primerjajo in določajo njihove ratinge. Lestvica je dostopna na naslovu: <http://computerchess.org.uk/>. Na tej lestvici je Crafty-21.6 dosegel rating 2647 ± 34 točk (01.04.2009). Rating je bil dosežen s pomočjo iger, ki so se igrale s časovno omejitvijo 40 minut za 40 potez na strojni opremi, ki je bila ekvivalentna Athlonu 64 X2 4600+ (2,4GHz). Iz rezultatov turnirja vidimo, da je najbolj ocenjen uglašen posameznik dosegel rating 2458 ± 37 točk. Igral je proti nasprotnikom, katerih povprečni rating je znašal 2391 točk, pri tem je dosegel 58% uspešnost in 19% iger je remiziral. Najboljši posameznik iz začetne populacije je dosegel rating 2309 ± 39 rating točk. Igral je proti nasprotnikom, katerih povprečni rating je znašal 2441 točk, pri tem je dosegel 35% uspešnost in 15% iger

je remiziral. Izboljšava najboljše ocenjenega posameznika uglasovanja glede na najboljšo ocenjenega posameznika iz začetne populacije znaša 149 rating točk. Iz rezultatov lahko vidimo tudi, da sta bila oba posameznika slabša od programa Crafty in boljša od programa GNU Chess. Iz tega lahko sklepamo, da smo uspešno uglasili program, katerega intervali uglasovanja so bili relativno dobro nastavljeni.

Izboljšava glede na igre, ki so bile odigrane proti programu Rybka, je bila 276 rating točk, v primeru iger, ki so igrane proti odprto-kodnemu programom, pa je bila 149 rating točk. Tudi dosežena igralna moč se je razlikovala. V prvem primeru je najboljšo ocenjeni uglaseni posameznik dosegel moč igranja 2202 rating točk, v drugem primeru 2458 rating točk. Te razlike so posledica različnih nasprotnikov, ki smo jih uporabljali za ocenjevanje posameznikov in različnih tipov turnirjev. V prvem je uporabljena nespremenjena globina iskanja (6), ki je bila enaka kot v procesu uglasovanja. V drugem primeru je bila uporabljena časovna omejitev, kjer je program ponavadi dosegel večje globine preiskovanja. Zato je bila verjetno izmerjena izboljšava v prvem primeru večja in dosežena moč igranja manjša, kot v drugem primeru.

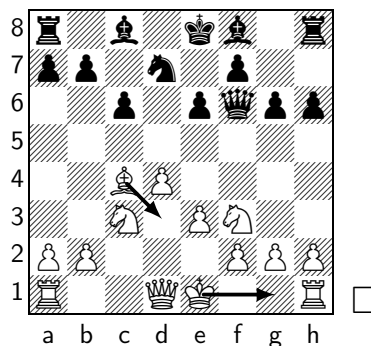
Moč uglasenega posameznika prikazujemo še s pomočjo analize ene izmed odigranih iger. Izbrali smo igro med uglasenim posameznikom in programom Crafty. To igro smo analizirali s pomočjo analize igre v grafičnem vmesniku Aquarium in programa Rybka 3. Po eni uri smo dobili naslednjo analizo:

Uglasen¹⁰₁ (2458) – Crafty-21.6 (2647)

40 potez v 5 minutah

Intel(R) Core(TM)2 CPU 6600 @ 2.40GHz

1 d4 d5 2 c4 c6 3 ♘c3 ♘f6 4 ♘f3 e6 5 ♙g5 h6 6 ♙xf6 ♚xf6 7 e3 ♘d7 8 ♙d3
dxc4 9 ♙xc4 g6 10 ♙d3

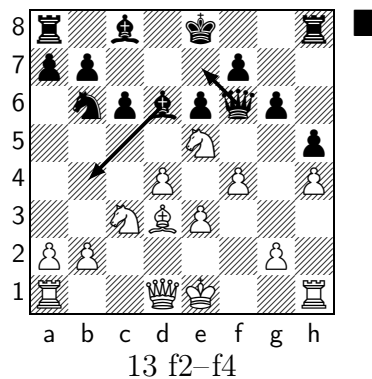


9...g7-g6

10 O-O ♖g7 11 ♖c2 O-O 12 ♘e4 ♗e7 13 ♙b3 e5 14 ♘c3 ♖h8 15 ♜ad1
b6 16 d5 ♙b7 17 h4 ♘c5 18 d6 ♗d7 19 h5 ♘xb3 20 axb3 g5 21 e4 ♜ad8
22 ♘e2 ♗e6 23 ♘g3 ♜xd6 24 ♘f5 ♜dd8 25 ♘h2 c5 26 ♘g4 a5 27 ♘ge3
♜fe8 28 ♘d5 ♜d7 29 ♘fe3 ♜ed8...1-0, Kramnik Vladimir 2807 - Piket
Jeroen 2646 , Monaco 2002 It "Melody Amber"(blindfold)

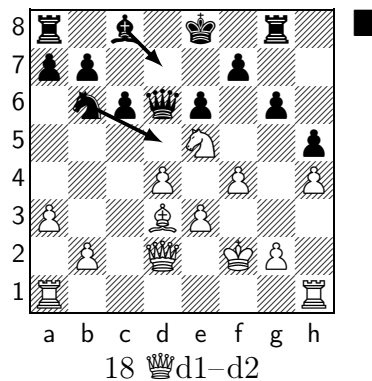
10 O-O ♖g7 11 b4 O-O 12 ♗b3 ♗e7 13 ♜ac1 b6 14 a4 ♙b7 15 ♜fd1
♜ab8 16 e4 ♜fd8 17 b5 c5 18 d5 ♘f8 19 a5 g5 20 h3 ♖h8 21 a6 ♙a8 22
♘e2 exd5 23 exd5 ♘g6 24 ♘g3 ♘f4 25 ♜e1 ♗f6 26 ♜cd1 ♜e8 27 ♜xe8+
♜xe8 28 d6 ♜d8 29 d7 ♘xh3+...1-0, Bacrot Etienne 2718 - Lautier Joel
2682 , Aix les Bains 2003 Ch France (play-off)

10...h5 11 h4 ♘b6 12 ♘e5 ♙d6 13 f4 +0.33



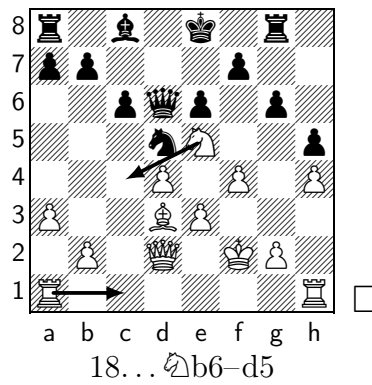
13...♙b4 14 a3 ♙xc3+ +0.37

13... ♗e7 +0.48 14 ♘e4 ♙b4+ 15 ♖f2 ♜g8 16 a3 ♙d6 17 ♘xd6+ ♗xd6 18
♗d2 ♘d5 +0.83



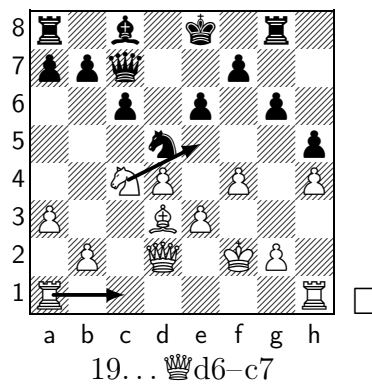
18... ♕d7 19 a4 ♖d5 20 ♜hc1 ♚b4 +0.56

19 ♗c4 +0.60



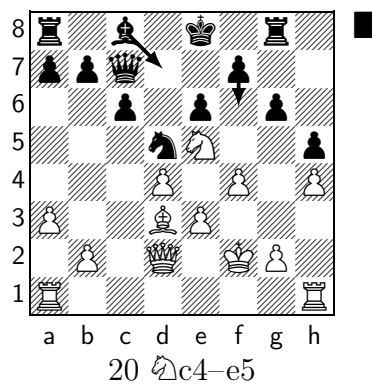
19 ♜ac1 ♕d7 +0.89

19... ♚c7 20 ♗e5 +0.18



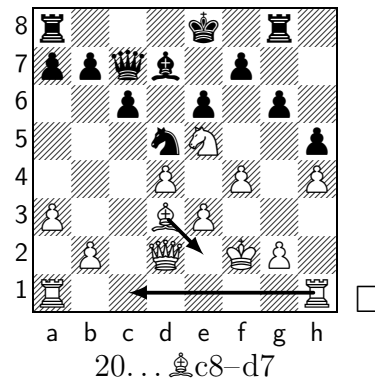
20 ♜ac1 ♕d7 +0.76

20... ♕d7 +0.73



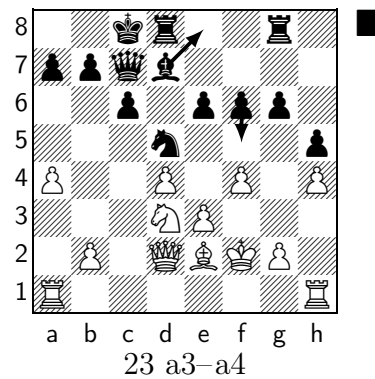
20... f6!? 21 ♖xg6 ♔f7 22 f5 +0.18

21 ♕e2 +0.18



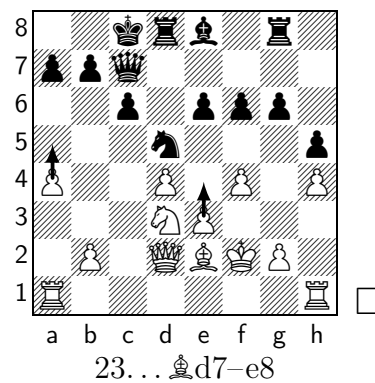
21 ♖hc1 !? f5 22 ♖c5 O-O-O +0.80

21... f6 22 ♖d3 O-O-O 23 a4 ♕e8 +0.52



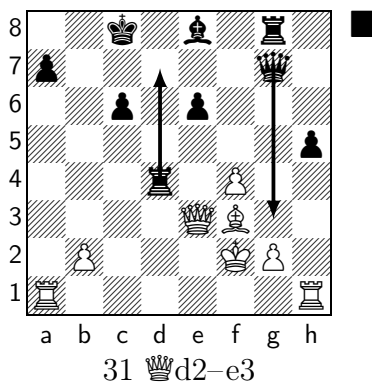
23... f5 24 ♖e5 ♖b8 +0.26

24 a5 -0.06



24 e4 !? ♖e7 25 ♘c5 ♙f7 26 ♚c4 ♖d6 27 ♗hd1 ♗gf8 28 ♜e3 g5 +0.60

24... g5 25 h×g5 f×g5 26 a6 g×f4 27 a×b7+ ♜×b7 28 ♘xf4 ♜g7 29 ♙f3 ♘xf4
30 exf4 ♗×d4 31 ♜e3 ♗d7 +0.39



31... ♜g3+!? 32 ♙e2 ♗b4 33 ♗a2 a6 34 ♗×h5 ♙×h5 -0.04

32 ♗hc1 ♜×b2+ +5.50 33 ♙f1 ♗e7 34 ♗ab1 ♜f6 35 ♜e4 e5 36 ♜b4 exf4 37
♜b8+ ♙d7 38 ♗×c6 ♜×c6 39 ♙×c6+ ♙×c6 40 ♜c8+ ♙d6 41 ♜d8+ ♙d7 42
♗d1+ ♙e6 43 ♜×g8+ ♙f5 44 g4+ f×g3 45 ♗d5+ ♙f4 46 ♜g5+ ♙e4 47 ♜×e7+
♙×d5 48 ♜×d7+1-0

V analizi so prikazane ključne pozicije, kjer program Rybka ocenjuje, da programa nista odigrala najboljših potez. Kot vidimo iz analize, je bila igra zelo dinamična. V potezi 31... d4d7 je program Crafty naredil ključno napako in omogočil najboljšemu uglasenemu posamezniku, da je uničil zaščito črnega kralja in zmagal. Boljša poteza bi bila 31... g7g3, ki bi omogočila enakopravno nadaljevanje igre.

4.4.3 Vpliv zgodovinskega mehanizma

Da bi ugotovili, kakšen vpliv ima vpeljan zgodovinski mehanizem na uglaševanje, smo uglaševanje ponovili brez njega. V algoritmu 3.10 smo odstranili vrstice 3, 14 in 16. Ker algoritem ni vseboval zgodovinske populacije, smo spremenili mutacijo. Uporabili smo strategijo DE/best/1. Algoritem je najboljšega posameznika, namesto iz zgodovinske populacije, vzel iz trenutne populacije. Kot v prejšnjih uglaševanjih, ko je bil krmilni parameter *igre* nastavljen na vrednost 10, smo ponovno zagnali 10 neodvisnih zagonov.

Ostale nastavitve so bile enake kot v prejšnjih uglaševanjih in so prikazane v tabeli 4.12.

Dobljene rezultate algoritma uglaševanja brez zgodovinskega mehanizma prikazuje tabela 5.4 (priloga str. 91) in v krajši obliki tabela 4.13. Oznaki Uglašen₁^{10*} in Začetni^{10*} predstavljata najbolje ocenjenega uglašene posameznika in najbolje ocenjenega posameznika iz začetne populacije, ko je bil algoritem zagnan brez zgodovinskega mehanizma. Iz rezultatov vidimo, da je bilo uglaševanje uspešno v vseh zagonih in da so izboljšave znašale od 68 do 216 rating točk. Najboljši rating je bil dosežen v 6 zagonu, 2046 ± 48 rating točk.

Primerjavo dobljenih rezultatov z zagoni, kjer je bil uporabljen zgodovinski mehanizem, prikazuje tabela 4.14. Na osnovi teh rezultatov vidimo, da vpeljan zgodovinski mehanizem bistveno izboljšuje uglaševanje parametrov na ozkih intervalih oz. takrat, ko poznamo približne vrednosti parametrov. Ta mehanizem omogoča algoritmu, da se izogiba prekomernemu učenju (ang. *Overfitting*), zmanjšuje šum pri ocenjevanju posameznikov in izboljša proces uglaševanja. Tako je naš algoritem, s pomočjo mehanizma nasprotij, dosežene ratinge v povprečju izboljšal za 155,2 rating točk. Izboljšave so bile v povprečju boljše za 105,7 rating točk. Vpeljan mehanizem nam je celo omogočil, da smo z manjšo vrednostjo krmilnega parametra *igre*, dosegli boljše rezultate. Z manj odigranimi igrami in manjšo časovno zahtevnostjo smo dosegli boljše posameznike. Ko je bil npr. parameter *igre* nastavljen na vrednost 2, je bilo odigranih 8000 iger. To je 5 krat manj v primerjavi z zagoni, ko je bil krmilni parameter nastavljen na vrednost 10. Torej se je časovna zahtevnost zmanjšala za 5 krat, povprečni doseženi

Tabela 4.12: Nastavitve algoritma, ki ni uporabljal zgodovinskega mehanizma

strategija	DE/best/1	Z_{vel}	20
N_p	20	Z_{vbriz}	4
D	190	<i>globina</i>	6
F	0,5	niti	2
C_r	0,9	predpomnilnik	16 MB
G	100	otvoritvena knjižnica	performance.bin
J_r	0,1	baze končnic	Scorpio bitbases
<i>igre</i>	10	predpomnilnik baz končnic	16 MB

Tabela 4.13: Rezultati uglaševanja brez uporabe zgodovinskega mehanizma

	1. zagon	2. zagon	3. zagon	4. zagon	5. zagon	6. zagon	7. zagon	8. zagon	9. zagon	10. zagon
Rybka ₂₃₀₀	2300±30	2300±34	2300±26	2300±32	2300±26	2300±26	2300±34	2300±33	2300±37	2300±25
Rybka ₂₁₀₀	2147±23	2139±25	2140±21	2123±23	2170±21	2150±21	2140±24	2103±23	2112±24	2143±21
Uglašen ₁ ^{10*}	1975±49	1907±52	2038±47	2002±46	2016±48	2046±48	1988±48	1952±49	1908±50	2041±47
Rybka ₁₉₀₀	2033±21	2035±21	1988±20	1949±20	1941±19	1959±19	1921±20	1980±20	1930±20	1980±19
Začetni ^{10*}	1875±54	1788±52	1896±51	1885±50	1800±54	1858±54	1920±49	1862±51	1823±51	1946±49
Rybka ₁₇₀₀	1732±21	1701±21	1780±21	1778±20	1686±23	1721±22	1753±19	1691±21	1699±20	1742±22
Izboljšava	100	119	142	117	216	188	68	90	85	95

Tabela 4.14: Vpliv zgodovinskega mehanizma

	Najboljši rating	Povprečni rating (σ)	Najboljša izboljšava	Povprečna izboljšava (σ)	Povprečni začetni rating (σ)	Število iger
Uglašen ₁ ¹⁰	2202±48	2142,5 (35,89)	372	227,7 (61,48)	1914,8 (66,15)	40000
Uglašen ₁ ⁵	2177±48	2132,8 (24,17)	432	245,9 (114,49)	1886,9 (100,86)	20000
Uglašen ₁ ²	2139±48	2087,3 (29,67)	276	176,9 (51,31)	1910,4 (47,2)	8000
Uglašen ₁ ^{10*}	2046±48	1987,3 (51,72)	216	122 (47,31)	1865,3 (50,58)	40000

rating pa je bil višji za 100 točk.

Dobljene rezultate (table 4.14) smo še statistično analizirali s pomočjo ne-parametričnih testov [25]. Uporabili smo Friedmanove in Iman-Devenportove teste, da bi ugotovili, ali obstajajo globalne razlike med rezultati. Za naš primer smo dobili kritične vrednosti 7,815 in 2,97 za $\alpha = 0,05$ ter naslednji statistični vrednosti (Friedmanova in Iman-Devenportova):

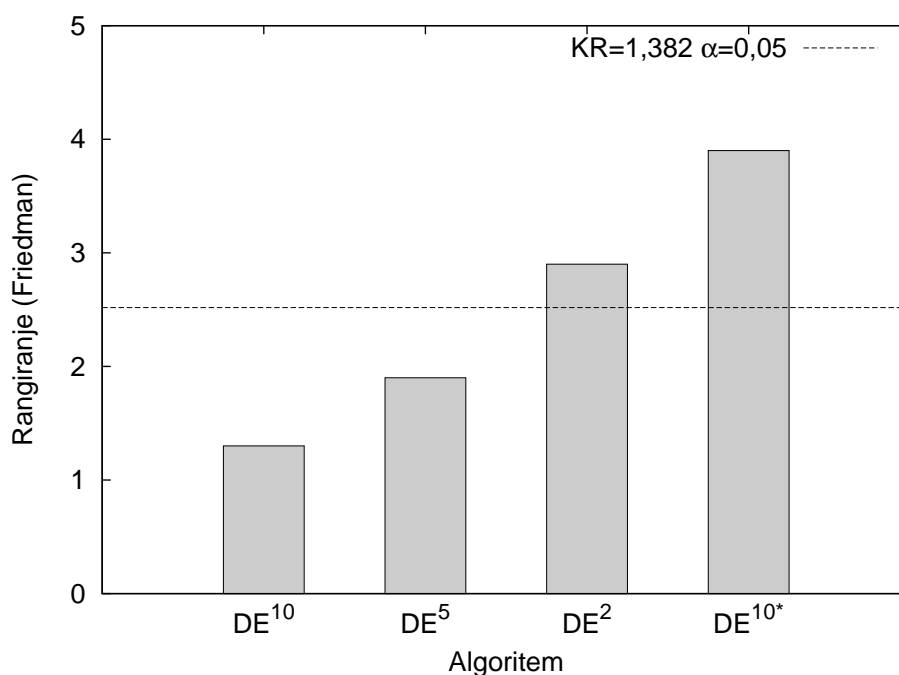
$$\chi_F^2 = 23,520, \quad F_F = 32,667. \quad (4.4)$$

Kritične vrednosti so manjše od pripadajočih statistik. To pomeni, da lahko uporabimo post-hoc teste in ugotovimo signifikantne razlike med algoritmi. Rezultate dobljene s pomočjo Friedmanovega testa prikazuje tabela 4.15. V tabeli so algoritmi rangirani in označeni s pomočjo oznak DE ter nadpisi. V nadpisih števila 2, 5 in 10 predstavljajo vrednost krmilnega parametra *igre* in * predstavlja algoritem, ki ne uporablja zgodovinskega mehanizma.

Za ugotavljanje signifikantnih razlik za referenčni algoritem DE^{10*} smo uporabili Bonferroni-Dunnov test. V našem primeru je bila kritična razlika $KR = 1,382$ pri $\alpha = 0,05$. Slika 4.6 prikazuje rangirane algoritme v obliki stolpcev in prag kritične razlike v obliki

Tabela 4.15: Rangiranje algoritmov s pomočjo Friedmanovega testa

Algoritem	Rang
DE ¹⁰	1,3
DE ⁵	1,9
DE ²	2,9
DE ^{10*}	3,8999999999999995

Slika 4.6: Bonferroni-Dunn test (referenčni algoritem: DE^{10*})

vodoravne črte. Ta črta predstavlja prag oz. razliko med doseženim rangom referenčnega algoritma in kritične razlike. Stolpci pod pragom predstavljajo algoritme, katerih učinkovitost je signifikantno boljša od referenčnega algoritma. To pomeni, da sta algoritma DE⁵ in DE¹⁰ signifikantno boljša od algoritma DE^{10*}.

Za izračun prilagojenih p -vrednosti smo uporabili večkratne primerjave med različnimi metodami [31]. Rezultate izračuna prikazuje tabela 4.16. Ta tabela v prvem stolpcu prikazuje hipoteze, da med določenima algoritmoma ni signifikantne razlike, v preostalih stolpcih pa prikazuje p -vrednosti in prilagojene p -vrednosti. S pomočjo neposredne primerjave stopnje signifikantnosti in prilagojenih p -vrednosti pa ugotovimo ali je do-

ločena hipoteza sprejeta oz. ovržena. Dodatno nam ta primerjava pove v kolikšni meri je hipoteza sprejeta oz. ovržena. V našem primeru za $\alpha = 0,5$ so vse metode zavrnila hipoteze 1 – 3. Za $\alpha = 0,1$ je metode Bergman-Hommel zavrnila hipoteze 1 – 4, vse ostale metode pa so zavrnila hipoteze 1 – 3.

Tabela 4.16: Prilagojene p -vrednosti

Hipoteza	Nepriprilagojena p -vrednost	Priprilagojena p -vrednost			
		Nemenyi	Holm	Shaffer	Bergman-Hommel
1 DE ¹⁰ nasp. DE ^{10*}	6,69e-6	4,01e-5	4,014e-5	4,01e-5	4,01e-5
2 DE ⁵ nasp. DE ^{10*}	5,32e-4	0,0032	0,0026	0,0016	0,0016
3 DE ¹⁰ nasp. DE ²	0,0056	0,0335	0,0223	0,0168	0,0168
4 DE ⁵ nasp. DE ²	0,0833	0,4996	0,2498	0,2498	0,0832
5 DE ² nasp. DE ^{10*}	0,0833	0,4996	0,2498	0,2498	0,1665
6 DE ¹⁰ nasp. DE ⁵	0,2987	1,7922	0,2987	0,2987	0,2987

Glede na analizo rezultatov lahko zaključimo, da zgodovinski mehanizem pomaga algoritmu pri izogibanju prekomernega učenja, zmanjšuje šum in izboljšuje učinkovitost uglaševanja. Algoritem z zgodovinskim mehanizmom se je izkazal kot signifikantno boljši. Signifikantno boljši je bil tudi v primeru, ko je imel nastavljen krmilni parameter *igre* na manjšo vrednost. To pomeni, da je algoritem s pomočjo manjšega števila iger dosegel signifikantno boljše rezultate.

4.5 Uglaševanje z evolucijskim algoritmom

Naš algoritem smo primerjali s podobnim evolucijskim algoritmom, ki je bil predstavljen v [30, 29]. Zaradi različnih šahovskih programov in nastavitvev krmilnih parametrov, je primerjava algoritmov za uglaševanje šahovskih programov težko opravilo. Različni šahovski programi vsebujejo različne ocenitvene funkcije in iskalne algoritme. Tako moč šahovskega programa, uporabljene podatkovne strukture, šahovsko znanje in oddaljenost parametrov od optimalnih vrednosti vplivajo na proces uglaševanja. Prav tako na uglaševanje vplivajo krmilni parametri algoritma uglaševanja, katerih ni možno nastaviti na optimalne vrednosti. Razlog temu je časovna zahtevnost uglaševanja.

Da bi naredili primerjavo čim bolj pravično, smo evlucijski algoritem implementirali glede na njegov opis v literaturi [30, 29] in z njim uglaševali naš šahovski program. Krmilne parametre algoritma smo nastavili prav tako glede na že omenjeno literaturo. V nadaljevanju tega podpoglavja podrobneje predstavimo implementirani evlucijski algoritem in rezultate njegovega uglaševanja, ki jih primerjamo z rezultati našega algoritma.

Implementirani evlucijski algoritem vsebuje eno populacijo P_g , ki jo sestavljajo starši in potomci. Za vsakega posameznika \vec{x}_i^g vsebuje populacija še vektor parametrov strategij \vec{s}_i^g in dosežene rezultate r_i^g . Opisano podatkovno strukturo prikazujejo neenačba in enačbe (4.5). To podatkovno strukturo uporablja evlucijski algoritem in vsebuje inicializacijo, mutacijo, ocenjevanje posameznikov in selekcijo, kot prikazuje algoritem 4.1. Podrobnejši opis naštetih sestavnih delov algoritma podajamo v naslednjih podpoglavjih.

$$\begin{aligned}
P_g &= \{ \{ \vec{x}_0^g, \vec{s}_0^g, r_0^g \}, \{ \vec{x}_1^g, \vec{s}_1^g, r_1^g \}, \dots, \{ \vec{x}_{starši-1}^g, \vec{s}_{starši-1}^g, r_{starši-1}^g \} \}, \dots, \\
&\quad \{ \vec{x}_{starši+potomci-1}^g, \vec{s}_{starši+potomci-1}^g, r_{starši+potomci-1}^g \} \\
\vec{x}_i^g &= \{ x_{i,0}^g, x_{i,1}^g, \dots, x_{i,D-1}^g \}, \quad \vec{s}_i^g = \{ s_{i,0}^g, s_{i,1}^g, \dots, s_{i,D-1}^g \}, \\
\vec{x}_{min} &= \{ x_{min,0}, x_{min,1}, \dots, x_{min,D-1} \}, \\
\vec{x}_{max} &= \{ x_{max,0}, x_{max,1}, \dots, x_{max,D-1} \}, \\
x_{min,j} &\leq x_{i,j}^g \leq x_{max,j}, \\
x_{i,j}^g, s_{i,j}^g &\in \mathbb{R}, \quad x_{min,j}, x_{max,j}, r_i^g \in \mathbb{Z}, \\
i &\in \{0, starši + potomci - 1\}, \quad j \in \{0, D - 1\}
\end{aligned} \tag{4.5}$$

4.5.1 Inicializacija

Pri inicializaciji se vrednosti parametrov starševskim posameznikom in parametrom strategij nastavijo naključno s pomočjo uniformne naključne porazdelitve in definiranih mej intervalov $x_{min,j}$ in $x_{max,j}$, kot prikazujeta enačbi (4.6) in algoritem 4.2.

$$\begin{aligned}
x_{i,j}^0 &= U(x_{j,low}, x_{j,high}) \\
s_{i,j}^0 &= U(0, (x_{j,high} - x_{j,low})/4) \\
i &\in \{0, starši - 1\}
\end{aligned} \tag{4.6}$$

Algoritem 4.1 Evolucijski algoritem

Oznake: P_g : populacija v g -ti generaciji, $starši$: število staršev v populaciji, $potomci$: število potomcev v populaciji, D : dimenzija problema, \vec{x}_{min} : vektor spodnjih mej parametrov, \vec{x}_{max} : vektor zgornjih mej parametrov, GEN : maksimalno število generacij, τ : krmilni parameter mutacije, ε : mejna vrednost parametrov strategije, $globina$: globina preiskovanja šahovskega programa in $igre$: krmilni parameter, ki določa število odigranih iger pri ocenjevanju posameznikov.

```

1: inicializacija( $P_g, \vec{x}_{min}, \vec{x}_{max}, starši, D$ );
2: for  $g = 0$  to  $GEN$  do
3:   mutacija( $P_g, \tau, \varepsilon, D$ );
4:   oceniPosameznike( $P_g, starši, potomci, globina, igre, D$ );
5:   selekcija( $P_g, D$ );
6: end for

```

Algoritem 4.2 Inicializacija v evolucijskem algoritmu

Oznake: i : indeks posameznika, j : indeks elementa vektorja, $starši$: število staršev v populaciji, D : dimenzija problema, $x_{i,j}^0$: j -ti element i -tega vektorja parametrov, $s_{i,j}^0$: j -ti element i -tega vektorja strategij, $x_{min,j}$: spodnja meja za j -te elemente vektorjev in $x_{max,j}$: zgornja meja za j -te elemente vektorjev.

```

1: for  $i = 0$  to  $starši$  do
2:   for  $j = 0$  to  $D$  do
3:      $x_{i,j}^0 = U(x_{j,low}, x_{j,high})$ ;
4:      $s_{i,j}^0 = U(0, (x_{j,high} - x_{j,low})/4)$ ;
5:   end for
6: end for

```

4.5.2 Mutacija

Mutacija ustvarja nove posameznike iz preživelih staršev. Iz vsakega starša se ustvari en potomec, kot prikazujeta enačba (4.7) in algoritem 4.3. Pri mutaciji vektorjev strategij je uporabljeno mejno pravilo. V primeru, ko element vektorja strategij dobi vrednost manjše od ε , se njegova vrednost nastavi na ε .

$$\begin{aligned}
 x_{i+starši,j}^g &= x_{i,j}^g + N(0, s_{i+starši,j}^g) \\
 s_{i+starši,j}^g &= \begin{cases} s_{i,j}^{g'}, & s_{i,j}^{g'} > \varepsilon \\ \varepsilon, & \text{drugače} \end{cases} \\
 s_{i,j}^{g'} &= s_{i,j}^{g-1} \cdot \tau \cdot \exp(N(0, 1)) \\
 i &\in \{0, starši-1\}
 \end{aligned} \tag{4.7}$$

Algoritem 4.3 Mutacija v evolucijskem algoritmu

Oznake: i : indeks posameznika, j : indeks elementa vektorja, $starši$: število staršev v populaciji, D : dimenzija problema, $x_{i,j}^g$: j -ti element i -tega vektorja parametrov, $s_{i,j}^g$: j -ti element i -tega vektorja strategij, τ krmilni parameter mutacije, $N(\mu, \sigma)$ Gaussova naključna porazdelitev s povprečno vrednostjo μ in standardno deviacijo σ , $x_{min,j}$: spodnja meja za j -te elemente vektorjev, $x_{max,j}$: zgornja meja za j -te elemente vektorjev in ε : spodnja meja parametrov strategij.

```

1: for  $i = 0$  to  $starši$  do
2:   for  $j = 0$  to  $D$  do
3:      $s_{i+starši,j}^g = s_{i,j}^g \cdot \tau \cdot \exp(N(0, 1))$ ;
4:     if  $s_{i+starši,j}^g < \varepsilon$  then
5:        $s_{i+starši,j}^g = \varepsilon$ ;
6:     end if
7:      $x_{i+starši,j}^g = x_{i,j}^g + N(0, s_{i+starši,j}^g)$ ;
8:   end for
9: end for

```

4.5.3 Ocenjevanje posameznikov

Da bi ocenili posameznike, le ti igrajo igre in zbirajo točke. Vsak posameznik odigra določeno število (*igre*) iger proti naključno izbranim posameznikom iz populacije (ne vključuje samega sebe). Pri tem polovico iger odigra z belimi in polovico iger s črnimi figurami. Vsota doseženih točk na koncu odigranih iger predstavlja oceno posameznika. Za zmago posameznik dobi 1 točko, za remi $\frac{1}{2}$ točke in za poraz 0 točk. Opisan način ocenjevanja posameznikov prikazujeta enačba (4.8) in algoritem 4.4.

$$\begin{aligned}
 r_i^g &= \sum_{k=0}^{igre/2-1} \text{igrajIgro}(\vec{x}_i^g, \vec{x}_n^g, \text{globina}) + \sum_{k=igre/2}^{igre-1} \left(1 - \text{igrajIgro}(\vec{x}_n^g, \vec{x}_i^g, \text{globina}) \right) \\
 n &= \text{rand}\{0, \text{starši} + \text{potomci} - 1\}, \quad n \neq i
 \end{aligned} \tag{4.8}$$

4.5.4 Selekcija

Selekcija glede na oceno posameznikov r_i^g določi, kateri posamezniki bodo preživeli v naslednjo generacijo. V implementiranem algoritmu v naslednjo generacijo preživijo le najboljše ocenjeni posamezniki. To pomeni, da se populacija uredi padajoče glede na oceno posameznikov in starši naslednje generacije so določeni.

Algoritem 4.4 Ocenjevanje posameznikov

Oznake: *starši*: število staršev v populaciji, *potomci*: število potomcev v populaciji, r_i : število doseženih točk i -tega posameznika, *igre*: krmilni parameter, ki določa število odigranih iger, n : nasprotnikov indeks, $\text{rand}\{0, \text{starši} + \text{potomci} - 1\}$: generator naključnih števil iz množice $\{0, \text{starši} + \text{potomci} - 1\}$, *globina*: globina preiskovanja šahovskega programa, \vec{x}_i^g, \vec{x}_n^g : posameznika iz populacije in g : številka trenutne generacije.

```

1: for  $i = 0$  to  $\text{starši} + \text{potomci} - 1$  do
2:    $r_i = 0$ ;
3:   for  $l = 0$  to igre do
4:      $n = i$ ;
5:     while  $n == i$  do
6:        $n = \text{rand}\{0, \text{starši} + \text{potomci} - 1\}$ ;
7:     end while
8:     if  $l \% 2 == 0$  then
9:        $r_i += \text{igrajIgro}(\vec{x}_i^g, \vec{x}_n^g, \text{globina})$ ;
10:    else
11:       $r_i += 1 - \text{igrajIgro}(\vec{x}_n^g, \vec{x}_i^g, \text{globina})$ ;
12:    end if
13:   end for
14: end for

```

4.5.5 Rezultati

Predstavljeni evlucijski algoritem smo uporabili za uglaševanje naše ocenitvene funkcije na podoben način kot v prejšnjih primerih. Zagnali smo 10 neodvisnih zagonov skozi 100 generacij. Vse nastavitve zagonov so prikazane v tabeli 4.17. Krmilni parameter *igre* smo nastavili na vrednost 10, ε in τ na 0,5, parametre šahovskega programa smo nastavili enako kot prej, število staršev in potomcev smo glede na literaturo [30, 29] nastavili na vrednost 10 in pri inicializaciji smo uporabili uniformno naključno porazdelitev.

Po končanih zagonih smo izračunali igralno moč posameznikov zadnje generacije in najbolje ocenjenega posameznika začetne populacije. Dobljene rezultate prikazuje tabela 5.5 in v krajši obliki tabela 4.18. Oznaki Uglašeni^{10#} in Začetni^{10#} predstavljata najbolje ocenjenega uglašenega posameznika in najbolje ocenjenega posameznika iz prve generacije. Iz prikazanih rezultatov vidimo, da je bil posameznik z največjo igralno močjo dosežen v četrtem zagonu. Njegova igralna moč je znašala 2150 ± 49 rating točk. Največja izboljšava v igralni moči je bila dosežena v šestem zagonu. Izboljšava je

Tabela 4.17: Nastavitve evolucijskega algoritma

starši	10	<i>globina</i>	6
potomci	10	nit	2
τ	0,5	hash	16 MB
<i>igre</i>	10	otvoritvena knjižnica	performance.bin
G	100	baze končnic	Scorpio bitbases
$x_{i,j}^0$	$U(x_{j,low}, x_{j,high})$	endgame hash	16 MB
$s_{i,j}^0$	$U(0, (x_{j,high} - x_{j,low})/4)$	ε	0,5

Tabela 4.18: Rezultati uglaševanja evolucijskega algoritma

	1. zagon	2. zagon	3. zagon	4. zagon	5. zagon	6. zagon	7. zagon	8. zagon	9. zagon	10. zagon
Rybka ₂₃₀₀	2300±24	2300±25	2300±24	2300±23	2300±22	2300±23	2300±28	2300±25	2300±23	2300±23
Rybka ₂₁₀₀	2148±20	2119±20	2145±20	2127±20	2127±19	2140±20	2113±21	2091±19	2151±20	2116±19
Uglašen ₁ ^{10#}	2075±49	2101±49	2061±47	2150±49	2140±48	2131±47	2048±48	2045±48	2140±49	2113±49
Rybka ₁₉₀₀	1961±19	1936±19	1909±19	1928±19	1970±19	1973±19	1943±19	1925±19	1916±20	1948±19
Začetni ₁ ^{10#}	1882±51	1837±51	1829±53	1924±49	1969±49	1717±58	1681±61	1986±49	1928±49	1923±49
Rybka ₁₇₀₀	1710±23	1727±22	1769±21	1717±24	1755±24	1759±23	1721±21	1712±22	1770±22	1756±22
Izboljšava	193	264	232	226	171	414	367	59	212	190

znašala 414 rating točk. Primerjavo rezultatov uporabljenega evolucijskega algoritma z rezultati našega algoritma prikazuje tabela 4.19. Vidimo, da je bilo v evolucijskem algoritmu odigrano enako število iger kot v primeru našega algoritma, ko je bil krmilni parameter *igre* nastavljen na vrednost 5. Naš algoritem je s to nastavitvijo dosegel boljše rezultate. Najboljši dosežen rating in povprečno dosežen rating je bil za 27 oz. 32,4 rating točk boljši. V primerjavi, ko je naš algoritem imel krmilni parameter *igre* nastavljen na vrednost 2, je odigral manj iger in dosegel slabše rezultate. V primeru, ko pa je imel nastavljen krmilni parameter *igre* na vrednost 10, je odigral več iger in tudi dosegel boljše rezultate.

Iz rezultatov vidimo, da je naš algoritem dosegel boljše rezultate od predstavljenega evolucijskega algoritma. Na osnovi opisane primerjave ni mogoče določiti, kateri od algoritmov je boljši. Za to opravilo bi morali algoritma preizkusiti z različnimi nastavitvami in ju primerjati z njihovimi optimalnimi nastavitvami. Zaradi časovne zahtevnosti problema je bilo to na žalost neizvedljivo.

Tabela 4.19: Primerjava evolucijskega algoritma z našim algoritmom

	Najboljši rating	Povprečni rating (σ)	Najboljša izboljšava	Povprečna izboljšava (σ)	Povprečni začetni rating (σ)	Število iger
Uglašen ₁ ¹⁰	2202±48	2142,5 (35,89)	372	227,7 (61,48)	1914,8 (66,15)	40000
Uglašen ₁ ⁵	2177±48	2132,8 (24,17)	432	245,9 (114,49)	1886,9 (100,86)	20000
Uglašen ₁ ²	2139±48	2087,3 (29,67)	276	176,9 (51,31)	1910,4 (47,2)	8000
Uglašen ₁ ^{10#}	2150±49	2100,4 (40,42)	414	232,8 (99,85)	1867,6 (102,38)	20000

4.6 Uglaševanje na širokih intervalih

Preverili smo tudi, kako se naš algoritem obnaša pri nekoliko širših intervalih uglaševanja. V ta namen smo definirali nove intervale, ki so prikazani v tabela 4.23. Ponovili smo 10 zagonov, katerih nastavitve so bile enake kot v primeru uglaševanja parametrov na ozkih intervalih in ko je bil krmilni parameter *igre* nastavljen na 10.

Tabela 4.20: Rezultati uglaševanja na širših intervalih

	1. zagon	2. zagon	3. zagon	4. zagon	5. zagon	6. zagon	7. zagon	8. zagon	9. zagon	10. zagon
Rybka ₂₃₀₀	2300±35	2300±28	2300±29	2300±27	2300±29	2300±28	2300±28	2300±26	2300±27	2300±26
Rybka ₂₁₀₀	2135±25	2159±22	2120±22	2088±20	2166±23	2162±22	2175±22	2166±21	2184±22	2172±22
Uglašen ₁ ¹⁰⁺	1901±49	2023±47	1993±48	2031±47	1976±48	1999±49	2030±47	2069±47	2068±47	2031±47
Rybka ₁₉₀₀	1964±20	1942±19	1918±19	1966±19	2014±20	1957±19	2016±20	1952±19	1956±19	2015±19
Začetni ¹⁰⁺	1189±179	1639±63	1500±77	1628±64	1239±158	1572±70	1509±83	1428±91	1444±91	1565±75
Rybka ₁₇₀₀	1783±20	1795±20	1740±20	1782±20	1772±21	1729±21	1754±21	1735±22	1750±21	1761±21
Izboljšava	712	384	493	493	737	427	521	641	624	466

Rezultate uglaševanja prikazuje tabela 5.6 in v krajši obliki tabela 4.20. Oznaki Uglašen₁¹⁰⁺ in Začetni¹⁰⁺ predstavljata najbolje uglašena posameznika in najbolj uglašena posameznika iz začetne populacije. Najbolje ocenjeni posameznik je bil dosežen v osmem zagonu, njegova igralna moč je znašala 2069 ± 47 rating točk. Največja izboljšava je bila dosežena v petem zagonu in sicer kar 737 rating točk.

Primerjavo dobljenih rezultatov s prejšnjimi zagoni prikazuje tabela 4.21. Najbolje ocenjeni posamezniki so dosegli manjšo igralno moč glede na prejšnje zagone. Rezultati so bili v povprečju za 130,4 rating točk slabši, če primerjamo rezultate dobljene z enakimi nastavitvami krmilnih parametrov. Tudi v primerjavi z nastavitvami, ko je algoritem odigral bistveno manj iger so rezultati slabši. Najbolje ocenjeni posamezniki

Tabela 4.21: Primerjava uglaševanja na ozkih in širokih intervalih

	Najboljši rating	Povprečni rating (σ)	Najboljša izboljšava	Povprečna izboljšava (σ)	Povprečni začetni rating (σ)	Število iger
Uglašen ₁ ¹⁰	2202±48	2142,5 (35,89)	372	227,7 (61,48)	1914,8 (66,15)	40000
Uglašen ₁ ⁵	2177±48	2132,8 (24,17)	432	245,9 (114,49)	1886,9 (100,86)	20000
Uglašen ₁ ²	2139±48	2087,3 (29,67)	276	176,9 (51,31)	1910,4 (47,2)	8000
Uglašen ₁ ¹⁰⁺	2069±47	2012,1 (49,13)	737	549,8 (121,13)	1471,5 (152,8)	40000

iz začetne populacije so prav tako dosegli nižje ratinge. Razlog slabšim rezultatom so širši intervali uglaševanja, na osnovi katerih so izbrane vrednosti parametrov bolj oddaljene od optimalnih. Bolj oddaljene vrednosti povzročajo večjo napako v ocenitveni funkciji in posledično imajo posamezniki manjšo igralno moč. Bolj, kot so oddaljene vrednosti parametrov od optimalnih, algoritem uglaševanja potrebuje več časa, da jih uglaši.

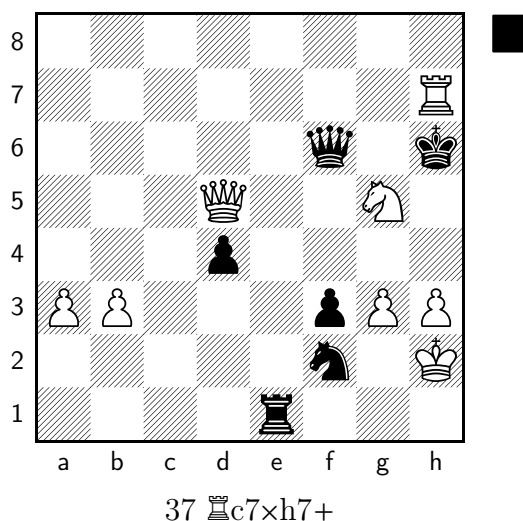
Vsi opisani zagoni in ocenjevanja posameznikov so časovno zahtevni in so trajali približno 6 mesecev. Časovna zahtevnost algoritma je odvisna od uporabljenih nastavitev. Opazili smo tudi, da velikost intervalov uglaševanja vpliva na časovno zahtevnost. Širši intervali uglaševanja povzročajo večje napake v ocenitveni funkciji. Te napake vplivajo tudi na iskalni algoritem, ki je včasih za doseganje globine 6 potreboval nekaj minut (uglaševanje brez ekspertnega znanja), kar je znatno več kot iskalni algoritem, ki vsebuje uglašeno ocenitveno funkcijo. S pomočjo uglašene ocenitvene funkcije se doseže globino 6 v nekaj milisekundah.

Tabela 4.22: Ozki intervali uglaševanja.

Ime	min	maks	x_j	Ime	min	maks	x_j	Ime	min	maks	x_j
PawnValueOpening	40	100	69	BishopCenterEndgame	-2	8	0	QueenLine[6]	-6	4	-3
KnightValueOpening	275	375	316	BishopBackRankOpening	0	20	8	QueenLine[7]	-8	2	1
BishopValueOpening	275	375	335	BishopDiagonalOpening	0	8	2	KingFile[0]	-2	8	5
RookValueOpening	450	550	538	RookFileOpening	0	6	2	KingFile[1]	-1	9	2
QueenValueOpening	925	1025	991	QueenCenterOpening	-5	5	1	KingFile[2]	-3	7	-2
PawnValueEndgame	60	120	88	QueenCenterEndgame	-1	9	0	KingFile[3]	-5	5	2
KnightValueEndgame	275	375	292	QueenBackRankOpening	0	10	4	KingFile[4]	-5	5	0
BishopValueEndgame	275	375	310	KingFileOpening	0	20	8	KingFile[5]	-3	7	4
RookValueEndgame	450	550	501	KingRankOpening	0	20	4	KingFile[6]	-1	9	5
QueenValueEndgame	925	1025	956	KingCenterEndgame	0	24	5	KingFile[7]	-2	8	4
BishopPairOpening	30	70	51	KnightOutpostMatrix[0][25]	-3	7	0	KingRank[0]	-4	6	-3
BishopPairEndgame	30	70	49	KnightOutpostMatrix[0][26]	0	10	1	KingRank[1]	-5	5	0
TrappedBishop	80	120	113	KnightOutpostMatrix[0][27]	5	15	9	KingRank[2]	-7	3	-3
BlockedBishop	30	70	41	KnightOutpostMatrix[0][28]	5	15	8	KingRank[3]	-8	2	2
BlockedRook	30	70	52	KnightOutpostMatrix[0][29]	0	10	3	KingRank[4]	-9	1	-4
KnightUnit	0	8	6	KnightOutpostMatrix[0][30]	-3	7	-1	KingRank[5]	-10	0	-8
BishopUnit	0	12	9	KnightOutpostMatrix[0][33]	-3	7	4	KingRank[6]	-11	-1	-4
RookUnit	0	14	6	KnightOutpostMatrix[0][34]	0	10	3	KingRank[7]	-12	-2	-6
KnightMobOpening	0	8	2	KnightOutpostMatrix[0][35]	5	15	6	KingLine[0]	-8	2	-4
KnightMobEndgame	0	8	2	KnightOutpostMatrix[0][36]	5	15	12	KingLine[1]	-6	4	1
BishopMobOpening	0	10	5	KnightOutpostMatrix[0][37]	0	10	9	KingLine[2]	-5	5	-4
BishopMobEndgame	0	10	6	KnightOutpostMatrix[0][38]	-3	7	2	KingLine[3]	-4	6	3
RookMobOpening	-2	6	0	KnightOutpostMatrix[0][42]	-1	9	6	KingLine[4]	-4	6	0
RookMobEndgame	0	8	5	KnightOutpostMatrix[0][43]	0	10	2	KingLine[5]	-5	5	-2
RookSemiOpenFileOpening	0	20	7	KnightOutpostMatrix[0][44]	0	10	2	KingLine[6]	-6	4	0
RookSemiOpenFileEndgame	0	20	3	KnightOutpostMatrix[0][45]	-1	9	0	KingLine[7]	-8	2	-3
RookOpenFileOpening	0	40	17	PawnFile[0]	-8	2	0	Bonus[3]	23	29	27
RookOpenFileEndgame	0	40	4	PawnFile[1]	-6	4	-2	Bonus[4]	69	85	73
RookSemiKingFileOpening	0	20	13	PawnFile[2]	-5	5	0	Bonus[5]	138	170	158
RookKingFileOpening	0	40	32	PawnFile[3]	-4	6	2	Bonus[6]	230	282	266
Rook7thOpening	0	40	17	PawnFile[4]	-4	6	-2	PawnRank[2]	0	2	0
Rook7thEndgame	0	80	51	PawnFile[5]	-5	5	0	PawnRank[3]	0	4	1
Queen7thOpening	0	20	10	PawnFile[6]	-6	4	0	PawnRank[4]	0	8	1
Queen7thEndgame	0	40	11	PawnFile[7]	-8	2	-3	PawnRank[5]	0	16	12
StormOpening	0	20	15	KnightLine[0]	-9	1	-6	PawnRank[6]	0	32	16
ShelterOpening	206	306	229	KnightLine[1]	-7	3	-4	PawnCenter[0]	0	20	11
KingAttackOpening	0	40	4	KnightLine[2]	-5	5	1	PawnCenter[1]	10	30	14
DoubledOpening	0	20	3	KnightLine[3]	-4	6	0	PawnCenter[2]	0	20	18
DoubledEndgame	0	40	8	KnightLine[4]	-4	6	2	KingAttackUnit[4]	0	2	0
IsolatedOpening	0	20	14	KnightLine[5]	-5	5	3	KingAttackUnit[5]	0	2	0
IsolatedOpeningOpen	0	40	24	KnightLine[6]	-7	3	-1	KingAttackUnit[6]	0	2	0
IsolatedEndgame	0	40	20	KnightLine[7]	-9	1	-6	KingAttackUnit[7]	0	2	0
BackwardOpening	0	16	9	BishopLine[0]	-8	2	0	KingAttackUnit[8]	0	4	0
BackwardOpeningOpen	0	32	12	BishopLine[1]	-6	4	0	KingAttackUnit[9]	0	4	1
BackwardEndgame	0	20	10	BishopLine[2]	-5	5	0	KingAttackUnit[10]	0	8	7
CandidateOpeningMin	0	10	7	BishopLine[3]	-4	6	2	KingAttackUnit[11]	0	8	4
CandidateOpeningMax	35	75	49	BishopLine[4]	-4	6	3	KingAttackWeight[0]	-1	1	0
CandidateEndgameMin	5	15	7	BishopLine[5]	-5	5	3	KingAttackWeight[1]	-1	1	0
CandidateEndgameMax	90	130	103	BishopLine[6]	-6	4	1	KingAttackWeight[2]	115	141	125
PassedOpeningMin	0	20	16	BishopLine[7]	-8	2	-1	KingAttackWeight[3]	172	212	173
PassedOpeningMax	50	90	87	RookFile[0]	-7	3	1	KingAttackWeight[4]	201	247	218
PassedEndgameMin	0	40	25	RookFile[1]	-6	4	1	KingAttackWeight[5]	215	265	255
PassedEndgameMax	110	170	136	RookFile[2]	-5	5	2	KingAttackWeight[6]	223	273	259
UnstoppablePasser	700	900	804	RookFile[3]	-4	6	0	KingAttackWeight[7]	226	278	259
AttackerDistance	0	10	6	RookFile[4]	-4	6	0	KingAttackWeight[8]	228	280	254
DefenderDistance	0	40	30	RookFile[5]	-5	5	2	KingAttackWeight[9]	229	281	266
PawnFileOpening	0	10	2	RookFile[6]	-6	4	1	KingAttackWeight[10]	230	282	257
PawnRankEndgame	0	7	1	RookFile[7]	-7	3	-2	KingAttackWeight[11]	230	282	244
KnightCenterOpening	0	10	2	QueenLine[0]	-8	2	-1	KingAttackWeight[12]	230	282	268
KnightRankOpening	0	10	2	QueenLine[1]	-6	4	1	KingAttackWeight[13]	230	282	279
KnightCenterEndgame	0	10	4	QueenLine[2]	-5	5	-1	KingAttackWeight[14]	230	282	263
KnightBackRankOpening	-5	5	3	QueenLine[3]	-4	6	2	KingAttackWeight[15]	230	282	264
KnightTrapped	80	120	92	QueenLine[4]	-4	6	0				
BishopCenterOpening	-3	7	2	QueenLine[5]	-5	5	0				

Tabela 4.23: Širši intervali uglaševanja.

Ime	min	maks	x_j	Ime	min	maks	x_j	Ime	min	maks	x_j
PawnValueOpening	50	150	122	BishopCenterEndgame	0	100	7	QueenLine[6]	-50	50	0
KnightValueOpening	250	350	296	BishopBackRankOpening	0	100	6	QueenLine[7]	-50	50	-5
BishopValueOpening	250	350	340	BishopDiagonalOpening	0	100	6	KingFile[0]	-50	50	0
RookValueOpening	450	550	488	RookFileOpening	0	100	8	KingFile[1]	-50	50	4
QueenValueOpening	900	1000	966	QueenCenterOpening	-50	50	0	KingFile[2]	-50	50	4
PawnValueEndgame	50	150	104	QueenCenterEndgame	-50	50	0	KingFile[3]	-50	50	2
KnightValueEndgame	250	350	274	QueenBackRankOpening	-50	50	0	KingFile[4]	-50	50	0
BishopValueEndgame	250	350	287	KingFileOpening	0	100	7	KingFile[5]	-50	50	0
RookValueEndgame	450	550	499	KingRankOpening	0	100	0	KingFile[6]	-50	50	2
QueenValueEndgame	900	1000	948	KingCenterEndgame	0	100	0	KingFile[7]	-50	50	0
BishopPairOpening	0	100	22	KnightOutpostMatrix[0][25]	0	100	0	KingRank[0]	-50	50	0
BishopPairEndgame	0	100	89	KnightOutpostMatrix[0][26]	0	100	12	KingRank[1]	-50	50	0
TrappedBishop	50	150	103	KnightOutpostMatrix[0][27]	0	100	3	KingRank[2]	-50	50	0
BlockedBishop	0	100	63	KnightOutpostMatrix[0][28]	0	100	5	KingRank[3]	-50	50	-5
BlockedRook	0	100	3	KnightOutpostMatrix[0][29]	0	100	9	KingRank[4]	-50	50	0
KnightUnit	0	100	1	KnightOutpostMatrix[0][30]	0	100	1	KingRank[5]	-50	50	-2
BishopUnit	0	100	0	KnightOutpostMatrix[0][33]	0	100	5	KingRank[6]	-50	50	0
RookUnit	0	100	5	KnightOutpostMatrix[0][34]	0	100	0	KingRank[7]	-50	50	0
KnightMobOpening	0	100	0	KnightOutpostMatrix[0][35]	0	100	5	KingLine[0]	-50	50	0
KnightMobEndgame	0	100	10	KnightOutpostMatrix[0][36]	0	100	12	KingLine[1]	-50	50	0
BishopMobOpening	0	100	6	KnightOutpostMatrix[0][37]	0	100	5	KingLine[2]	-50	50	0
BishopMobEndgame	0	100	7	KnightOutpostMatrix[0][38]	0	100	9	KingLine[3]	-50	50	0
RookMobOpening	0	100	4	KnightOutpostMatrix[0][42]	0	100	6	KingLine[4]	-50	50	0
RookMobEndgame	0	100	7	KnightOutpostMatrix[0][43]	0	100	7	KingLine[5]	-50	50	0
RookSemiOpenFileOpening	0	100	1	KnightOutpostMatrix[0][44]	0	100	10	KingLine[6]	-50	50	6
RookSemiOpenFileEndgame	0	100	22	KnightOutpostMatrix[0][45]	0	100	2	KingLine[7]	-50	50	4
RookOpenFileOpening	0	100	37	PawnFile[0]	-50	50	0	Bonus[3]	0	100	41
RookOpenFileEndgame	0	100	24	PawnFile[1]	-50	50	0	Bonus[4]	0	100	32
RookSemiKingFileOpening	0	100	15	PawnFile[2]	-50	50	0	Bonus[5]	100	200	174
RookKingFileOpening	0	100	11	PawnFile[3]	-50	50	0	Bonus[6]	200	300	232
Rook7thOpening	0	100	14	PawnFile[4]	-50	50	0	PawnRank[2]	0	100	9
Rook7thEndgame	0	100	17	PawnFile[5]	-50	50	0	PawnRank[3]	0	100	11
Queen7thOpening	0	100	12	PawnFile[6]	-50	50	3	PawnRank[4]	0	100	1
Queen7thEndgame	0	100	25	PawnFile[7]	-50	50	-3	PawnRank[5]	0	100	2
StormOpening	0	100	16	KnightLine[0]	-50	50	-6	PawnRank[6]	0	100	7
ShelterOpening	200	300	284	KnightLine[1]	-50	50	0	PawnCenter[0]	0	100	30
KingAttackOpening	0	100	9	KnightLine[2]	-50	50	0	PawnCenter[1]	0	100	27
DoubledOpening	0	100	31	KnightLine[3]	-50	50	0	PawnCenter[2]	0	100	39
DoubledEndgame	0	100	0	KnightLine[4]	-50	50	0	KingAttackUnit[4]	-50	50	2
IsolatedOpening	0	100	2	KnightLine[5]	-50	50	0	KingAttackUnit[5]	-50	50	3
IsolatedOpeningOpen	0	100	18	KnightLine[6]	-50	50	0	KingAttackUnit[6]	-50	50	0
IsolatedEndgame	0	100	10	KnightLine[7]	-50	50	1	KingAttackUnit[7]	-50	50	7
BackwardOpening	0	100	36	BishopLine[0]	-50	50	0	KingAttackUnit[8]	-50	50	8
BackwardOpeningOpen	0	100	9	BishopLine[1]	-50	50	0	KingAttackUnit[9]	-50	50	5
BackwardEndgame	0	100	5	BishopLine[2]	-50	50	0	KingAttackUnit[10]	-50	50	2
CandidateOpeningMin	0	100	31	BishopLine[3]	-50	50	-1	KingAttackUnit[11]	-50	50	3
CandidateOpeningMax	0	100	78	BishopLine[4]	-50	50	0	KingAttackWeight[0]	-50	50	5
CandidateEndgameMin	0	100	31	BishopLine[5]	-50	50	0	KingAttackWeight[1]	-50	50	17
CandidateEndgameMax	50	150	59	BishopLine[6]	-50	50	0	KingAttackWeight[2]	100	200	151
PassedOpeningMin	0	100	9	BishopLine[7]	-50	50	-3	KingAttackWeight[3]	100	200	124
PassedOpeningMax	0	100	88	RookFile[0]	-50	50	0	KingAttackWeight[4]	200	300	221
PassedEndgameMin	0	100	16	RookFile[1]	-50	50	0	KingAttackWeight[5]	200	300	233
PassedEndgameMax	50	150	61	RookFile[2]	-50	50	0	KingAttackWeight[6]	200	300	242
UnstoppablePasser	750	850	799	RookFile[3]	-50	50	0	KingAttackWeight[7]	200	300	241
AttackerDistance	0	100	11	RookFile[4]	-50	50	0	KingAttackWeight[8]	200	300	271
DefenderDistance	0	100	24	RookFile[5]	-50	50	2	KingAttackWeight[9]	200	300	246
PawnFileOpening	0	100	5	RookFile[6]	-50	50	0	KingAttackWeight[10]	200	300	241
PawnRankEndgame	0	100	0	RookFile[7]	-50	50	0	KingAttackWeight[11]	200	300	223
KnightCenterOpening	0	100	3	QueenLine[0]	-50	50	-1	KingAttackWeight[12]	200	300	210
KnightRankOpening	0	100	4	QueenLine[1]	-50	50	4	KingAttackWeight[13]	200	300	250
KnightCenterEndgame	0	100	2	QueenLine[2]	-50	50	0	KingAttackWeight[14]	200	300	256
KnightBackRankOpening	0	100	13	QueenLine[3]	-50	50	0	KingAttackWeight[15]	200	300	271
KnightTrapped	50	150	119	QueenLine[4]	-50	50	0				
BishopCenterOpening	0	100	0	QueenLine[5]	-50	50	0				



Napad je duša šaha. Zmaga je cvet in sad napada. (Milan Vidmar)

5 Zaključek

V delu predstavimo algoritem za uglaševanje šahovske ocenitvene funkcije. Ta algoritem temelji na algoritmu diferencialne evolucije, ki uporablja mehanizem nasprotij. Da bi izboljšali proces uglaševanja, algoritem nadgradimo z zgodovinskim mehanizmom. Ta mehanizem vsebuje dodatno populacijo, potencialno dobrih posameznikov. Čeprav potencialno dobri posamezniki v določeni generaciji izumrejo, jih zgodovinski mehanizem s pomočjo dodatne populacije lahko vrne nazaj v proces uglaševanja. Tako potencialno dobre posameznike uglašujemo skozi več različnih okoljih oz. populacij. Posledica tega je, da ti posamezniki odigrajo več iger proti različnim posameznikom in njihova ocena igralne moči vsebuje manjšo napako. S tem zmanjšamo šum, možnost prekomernega učenja posameznikov in izboljšujemo celoten proces uglaševanja.

Predstavljeni algoritem smo preizkusili s pomočjo uglaševanja ocenitvene funkcije našega šahovskega programa BBChess. Uglaševali smo ga s pomočjo različnih intervalov uglaševanja. Najprej smo ga uglaševali na ekstremno širokih intervalih. Tako smo preizkusili, ali lahko algoritem poišče vrednosti parametrov, za katere niti približno ne

vemo, kje so njihove optimalne vrednosti. Ugotovili smo, da algoritem najde vrednosti parametrov, ki ustrezajo razmerju, znanemu iz teorije šaha.

Algoritem smo preizkusili še z uglasenjem ocenitvene funkcije s pomočjo ozkih intervalov uglasenja. V procesu razvoja šahovskih programov poznajo razvijalci približne vrednosti parametrov. Z nastavitvijo intervalov okoli približnih vrednosti in uglasenjem lahko izboljšajo igralno moč programov. Tako smo tudi mi določili ozke intervale uglasenja in našo ocenitveno funkcijo uglasili z različnim številom odigranih iger potrebnih za ocenjevanje posameznikov. Iz rezultatov smo ugotovili korelacijo, da več iger omogoča boljše uglasenje. Razlog temu je, da večje število iger zmanjšuje šum oz. možnost, da slabši posamezniki zmagajo v dvobojih in preživijo v naslednje generacije.

Da bi pokazali vpliv zgodovinskega mehanizma, smo uglasenje ponovili še z algoritmom brez zgodovinskega mehanizma. Ta je v vseh zagonih uspešno uglašil ocenitveno funkcijo oz. izboljšal njeno kvaliteto. S tem smo potrdili prvo hipotezo, da je algoritem diferencialne evolucije, ki uporablja mehanizem nasprotij, primeren za uglasenje šahovske ocenitvene funkcije. Dobljene rezultate smo nato primerjali še z rezultati algoritma, ki je uporabljal zgodovinski mehanizem. Izkazalo se je, da zgodovinski mehanizem bistveno izboljša proces uglasenja. Z zgodovinskim mehanizmom, in s precej manj odigranimi igrami, je algoritem dosegel boljše rezultate. Razlog temu je, da so potencialno dobri posamezniki odigrali več iger nasproti različnim posameznikom. Tako se je zmanjšal šum pri njihovem ocenjevanju in tudi možnost prekomernega učenja. S tem smo potrdili tudi drugo hipotezo, da zgodovinski mehanizem izboljšuje učinkovitost algoritma diferencialne evolucije, ki uporablja mehanizem nasprotij.

Naš algoritem smo primerjali z znanim evolucijskim algoritmom za uglasenje šahovske ocenitvene funkcije [30] tako, da smo ga implementirali in uglasili našo ocenitveno funkcijo s pomočjo ozkih intervalov uglasenja. Dobljeni rezultati kažejo, da je naš algoritem dosegel boljše rezultate. Zaradi časovne zahtevnosti problema ni možno narediti popolne primerjave algoritmov. Zato lahko trdimo, da je naš algoritem primerljiv z znanim evolucijskim algoritmom iz literature [30].

Učinkovitost našega algoritma smo preizkusili še s pomočjo nekoliko širših intervalov uglaševanja. Tako smo povečali iskalni prostor, napake v ocenitveni funkciji in možnost za izboljšave. To so potrdili tudi rezultati. Posamezniki začetne populacije so dosegli manjše igralne moči, izboljšave uglaševanja so se povečale in ocene najbolj ocenjenih posameznikov so se zmanjšale. Tako lahko vidimo, da je bil naš algoritem uspešen tudi v primeru, ko so bili intervali nekoliko širši. S tem smo potrdili tretjo hipotezo, da algoritem diferencialne evolucije, ki uporablja mehanizem nasprotij in zgodovinski mehanizem, omogoča učinkovito uglaševanje parametrov tako na širokih kot na ozkih intervalih uglaševanja.

Predstavljeni algoritem bi bilo zanimivo preizkusiti še na drugih odprto-kodnih šahovskih programih, podobnih igrah in problemih. Prav tako je možno vpeljani zgodovinski mehanizem uporabiti tudi v drugih evlucijskih algoritmih in algoritmih za uglaševanje šahovske ocenitvene funkcije.

Literatura

- [1] T. S. Anantharaman. *A Statistical Study of Selective Min-Max Search in Computer Chess*. PhD thesis, Carnegie Mellon University Pittsburgh, U.S.A., 1990.
- [2] T. S. Anantharaman. Evaluation Tuning for Computer Chess: Linear Discriminant Methods. *ICCA Journal*, 20(4):224–242, 1997.
- [3] George W. Atkinson. *Chess and Machine Intuition*. Intellect Books, 1998.
- [4] Jonathan Baxter, Andrew Tridgell in Lex Weaver. Experiments in Parameter Learning Using Temporal Differences. *International Computer Chess Association Journal*, 21(2):84–99, 1998.
- [5] Jonathan Baxter, Andrew Tridgell in Lex Weaver. Learning to Play Chess Using Temporal Differences. *Machine Learning*, 40(3):243–263, 2000.
- [6] D. F. Beal in M. C. Smith. Learning Piece values Using Temporal Differences. *Journal of The International Computer Chess Association*, 20(3):147–151, 1997.
- [7] D. F. Beal in M. C. Smith. First Results from Using Temporal Difference Learning in Shogi. V: H. J. van den Herik in H. Iida, uredniki, *Computers and Games*, str. 113–125. Lecture Notes in Computer Science 1558, Springer-Verlag, Berlin, Germany, 1999.
- [8] D. F. Beal in M. C. Smith. Learning Piece-Square Values using Temporal Differences. *ICCA Journal*, 22(4):223–235, 1999.
- [9] A. Bernstein, M. de V. Roberts, T. Arbuckle in M. S. Belsky. A chess playing program for the IBM 704. V: *Proceedings of the 1958 Western Joint Computer Conference*, str. 157–159, 1958.

- [10] B. Bošković, S. Greiner, J. Brest in V. Žumer. The Representation of Chess Game. V: *Proceedings of the 27th International Conference on Information Technology Interfaces*, str. 381–386, 2005.
- [11] B. Bošković, S. Greiner, J. Brest in V. Žumer. A Differential Evolution for the Tuning of a Chess Evaluation Function. V: *The 2006 IEEE Congress on Evolutionary Computation CEC2006*, str. 6742–6747. IEEE Press, 2006.
- [12] B. Bošković, S. Greiner, J. Brest, A. Zamuda in V. Žumer. An Adaptive Differential Evolution Algorithm with Opposition-Based Mechanisms, Applied to the Tuning of a Chess Program. V: Uday K. Chakraborty, urednik, *Advances in Differential Evolution, Studies in Computational Intelligence*, Springer, June 2008.
- [13] J. Brest, B. Bošković, S. Greiner, V. Žumer in M. Sepesy Maučec. Performance comparison of self-adaptive and adaptive differential evolution algorithms. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 11(7):617–629, 2007.
- [14] J. Brest, S. Greiner, B. Bošković, M. Mernik in V. Žumer. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657, 2006.
- [15] J. Brest in M. Sepesy Maučec. Population Size Reduction for the Differential Evolution Algorithm. *Applied Intelligence*, 29(3):228–247, 2008.
- [16] D. M. Breuker, J. W. H. M. Uiterwijk in H. J. Van Den Herik. Replacement schemes for transposition tables. *ICCA Journal*, 17:183–193, 1994.
- [17] M. Buro. Statistical Feature Combination for the Evaluation of Game Positions. *Journal of Artificial Intelligence Research* 3, str. 373–382, 1995.
- [18] M. Buro. From Simple Features to Sophisticated Evaluation Functions. V: H. J. van den Herik in H. Iida, uredniki, *Computers and Games*, str. 126–145. Lecture Notes in Computer Science 1558, Springer-Verlag, 1999.
- [19] Stuart Russell in Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, 2003.

- [20] K. Chellapilla in D. B. Fogel. Evolving Neural Networks to Play Checkers Without Relying on Expert Knowledge. *IEEE Transactions on Neural Networks*, 10(6):1382–1391, 1999.
- [21] K. Chellapilla in D. B. Fogel. Evolving an Expert Checkers Playing Program Without Using Human Expertise. *IEEE Transactions on Evolutionary Computation*, 5(4):422–428, 2001.
- [22] David Clark. Deep thoughts on Deep Blue. *IEEE Expert: Intelligent Systems and Their Applications*, 12(4):31, 1997.
- [23] Swagatam Das, Amit Konar, Uday K. Chakraborty in Ajith Abraham. Differential evolution with a neighborhood based mutation operator: a comparative study. *IEEE Transactions on Evolutionary Computation*, 13(3):526–553, 2009.
- [24] David B. Fogel, Timothy J. Hays, Sarah L. Hahn in James Quon. The Blondie25 Chess Program Competes Against Fritz 8.0 and a Human Chess Master. V: Sushil J. Louis in Graham Kendall, uredniki, *The 2006 IEEE Symposium on Computational Intelligence and Games*, str. 230–235. IEEE, 2006.
- [25] J. Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [26] Vitaliy Feoktistov. *Differential Evolution: In Search of Solutions (Springer Optimization and Its Applications)*. Springer-Verlag New York, Inc, Secaucus, NJ, USA, 2006.
- [27] Harald Fietz. Beyond the 3000 Elo barrier A glance behind the scenes of the Rybka chess engine. *Chess*, str. 18 – 21, 2007.
- [28] David B. Fogel. *Blondie24, Playing at the Edge of AI*. Morgan Kaufmann, 2002.
- [29] David B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Wiley-IEEE Press, third edition, 2006.
- [30] David B. Fogel, Timothy J. Hays, Sarah L. Hahn in James Quon. A Self-Learning Evolutionary Chess Program. *Proceedings of the IEEE*, 92(12):1947–1954, 2004.

- [31] S. García in F. Herrera. An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.
- [32] D. Gomboc, M. Buro in T. A. Marsland. Tuning evaluation functions by maximizing concordance. *Theoretical Computer Science*, 349(2):202–229, 2005.
- [33] R. Greenblatt, D. Eastlake in S. Crocker. The Greenblatt Chess Program. V: *Proceedings of the November 14-16, 1967, fall joint computer conference*, str. 801–810, 1967.
- [34] Ernst A. Heinz. How DarkThought plays chess. *ICCA Journal*, 20(3):166–176, 1997.
- [35] Ernst A. Heinz. *Scalable Search in Computer Chess: Algorithmic Enhancements and Experiments at High Search Depths (Computational Intelligence)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [36] F.-H. Hsu, T. S. Anantharaman, M. S. Campbell in A. Nowatzyk. Deep Thought. V: T. A. Marsland in J. Schaeffer, uredniki, *Computers, Chess and Cognition*, poglavje 5, str. 55–78. Springer-Verlag, 1990.
- [37] David R. Hunter. MM algorithms for generalized Bradley-Terry models. *Annals of Statistics*, 32(1):384–406, 2004.
- [38] Johannes Fürnkranz. Machine Learning in Games: A Survey. V: J. Fürnkranz in M. Kubat, uredniki, *Machines that Learn to Play Games*, str. 11–59. Nova Science Publishers, Inc., 2001.
- [39] G. Kendall in G. Whitwell. An Evolutionary Approach for the Tuning of a Chess Evaluation Function Using Population Dynamics. V: *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, str. 995–1002, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 2001. IEEE Press.
- [40] Marc Lacrosse. Private communication. 2008.
- [41] R. Levinson in R. Snyder. Adaptive Pattern-Oriented Chess. *AAAI*, str. 601–606, 1991.

- [42] D. N. L. Levy in M. Newborn. *How Computers Play Chess*. New York: Computer Science Press, 1991.
- [43] T. A. Marsland. Evaluation Function Factors. *International Computer Chess Association Journal*, 8(2):47–57, 1985.
- [44] P. Mysliwietz. *Konstruktion und Optimierung von Bewertungsfunktionen beim Schach*. PhD thesis, University of Paderborn, Germany, 1994.
- [45] H. Nasreddine, H.S. Poh in G. Kendall. Using an Evolutionary Algorithm for the Tuning of a Chess Evaluation Function Based on a Dynamic Boundary Strategy. V: *Proceedings of 2006 IEEE International Conference on Cybernetics and Intelligent Systems (CIS2006)*, str. 1–6, 2006.
- [46] A. Newell, J. C. Shaw in H. A. Simon. Chess Playing Programs and the Problem of Complexity. *IBM Journal of Research and Development*, 2(4):320–325, 1958.
- [47] K. V. Price, R. M. Storn in J. A. Lampinen. *Differential Evolution, A Practical Approach to Global Optimization*. Springer, 2005.
- [48] A. K. Qin, V. L. Huang in P. N. Suganthan. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation*, 13(2):398–417, 2009.
- [49] S. Rahnamayan, H. R. Tizhoosh in M. M. A. Salama. Opposition-Based Differential Evolution Algorithms. V: *The 2006 IEEE Congress on Evolutionary Computation CEC2006*, str. 7363–7370, 2006.
- [50] S. Rahnamayan, H. R. Tizhoosh in M. M. A. Salama. Opposition-Based Differential Evolution. *IEEE Transactions on Evolutionary Computation*, 12(1):64–79, 2008.
- [51] D. Rasmussen. Parallel chess searching and bitboards. Master’s thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2004. Supervised by Prof. Jens Clausen.

- [52] A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3):211–229, 1959.
- [53] A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. II—Recent Progress. *IBM Journal of Research and Development*, 11(6):601–617, 1967.
- [54] J. Schaeffer, M. Hlynka in V. Jussila. Temporal Difference Learning Applied to a High-Performance Game-Playing Program. V: *International Joint Conference on Artificial Intelligence*, str. 529–534, 2001.
- [55] C. Shannon. Programming a computer for playing chess. *Philosophical Magazine*, 41(4):256, 1950.
- [56] R. Storn in K. Price. Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Berkeley, CA, 1995.
- [57] R. Storn in K. Price. Differential Evolution - A Simple and Efficient Heuristic for Global Optimisation Over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.
- [58] D. D. Swade. Redeeming Charles Babbage’s Mechanical Computer. *Scientific American Magazine*, 268(2):86–91, 1993.
- [59] G. Tesauro. Connectionist learning of expert preferences by comparison training. V: D. Touretzky, urednik, *Advances in neural information processing systems 1*, str. 99–106. Morgan Kaufmann, 1989.
- [60] G. Tesauro. Temporal Difference Learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.
- [61] G. Tesauro. Comparison Training of Chess Evaluation Functions. V: J. F. urnkranz in M. Kubat, uredniki, *Machines that Learn to Play Games*, str. 117–130. Nova Science Publishers, 2001.

-
- [62] Sebastian Thrun. Learning To Play the Game of Chess. V: G. Tesauero, D. Touretzky in T. Leen, uredniki, *Advances in Neural Information Processing Systems* 7, str. 1069–1076. The MIT Press, Cambridge, MA, 1995.
- [63] H. R. Tizhoosh. Opposition-Based Learning: A New Scheme for Machine Intelligence. V: *Proceedings of International Conference on Computational Intelligence for Modelling Control and Automation - CIMCA2005*, str. 695–701, Dunaj, Avstrija, 2005.
- [64] M. van der Meulen. Weight Assessment in Evaluation Functions. V: D. F. Beal, urednik, *Advances in Computer Chess 5*, str. 81–90. Elsevier Science Publishers, Amsterdam, 1989.
- [65] Jingqiao Zhang in Arthur C. Sanderson. JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5):945–958, 2009.

Priloga

Rezultati poskusov uglaševanja ocenitvene funkcije na ozkih intervalih. Podrobnejši opis poskusa je na str. 45.

Tabela 5.1: Rezultati uglaševanja na ozkih intervalih ($igre = 10$)

(a) 1. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	21	21	1050	81%	2038	12%
2	Rybka ₂₁₀₀	2157	19	19	1050	65%	2038	13%
3	Uglašen ₁ ¹⁰	2145	48	48	200	60%	2037	13%
4	Uglašen ₂ ¹⁰	2140	48	48	200	60%	2037	13%
5	Uglašen ₃ ¹⁰	2103	48	48	200	57%	2037	11%
6	Uglašen ₄ ¹⁰	2098	49	49	200	56%	2037	6%
7	Uglašen ₅ ¹⁰	2092	47	47	200	55%	2037	14%
8	Uglašen ₆ ¹⁰	2088	49	49	200	55%	2037	7%
9	Uglašen ₇ ¹⁰	2066	49	49	200	52%	2037	8%
10	Uglašen ₈ ¹⁰	2059	47	47	200	52%	2037	15%
11	Uglašen ₉ ¹⁰	2039	49	49	200	51%	2037	13%
12	Uglašen ₁₀ ¹⁰	2037	46	46	200	50%	2037	19%
13	Uglašen ₁₁ ¹⁰	2027	46	46	200	49%	2037	23%
14	Uglašen ₁₂ ¹⁰	2023	48	48	200	48%	2037	11%
15	Uglašen ₁₃ ¹⁰	2022	47	47	200	49%	2037	11%
16	Uglašen ₁₄ ¹⁰	2014	49	49	200	47%	2037	10%
17	Uglašen ₁₅ ¹⁰	2010	49	49	200	47%	2037	6%
18	Uglašen ₁₆ ¹⁰	2006	48	48	200	46%	2037	19%
19	Uglašen ₁₇ ¹⁰	1986	49	49	200	46%	2037	13%
20	Uglašen ₁₈ ¹⁰	1984	49	49	200	44%	2037	7%
21	Uglašen ₁₉ ¹⁰	1982	49	49	200	44%	2037	16%
22	Uglašen ₂₀ ¹⁰	1982	49	49	200	44%	2037	8%
23	Rybka ₁₉₀₀	1957	19	19	1050	40%	2038	13%
24	Začetni ¹⁰	1902	49	49	200	35%	2037	14%
25	Rybka ₁₇₀₀	1733	23	23	1050	15%	2038	10%

(b) 2. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	21	21	1050	79%	2061	14%
2	Rybka ₂₁₀₀	2163	20	20	1050	63%	2061	12%
3	Uglašen ₁ ¹⁰	2151	48	48	200	61%	2048	10%
4	Uglašen ₂ ¹⁰	2150	48	48	200	61%	2048	11%
5	Uglašen ₃ ¹⁰	2139	47	47	200	58%	2048	20%
6	Uglašen ₄ ¹⁰	2130	48	48	200	59%	2048	13%
7	Uglašen ₅ ¹⁰	2102	47	47	200	56%	2048	14%
8	Uglašen ₆ ¹⁰	2092	48	48	200	55%	2048	9%
9	Uglašen ₇ ¹⁰	2091	47	47	200	55%	2048	17%
10	Uglašen ₈ ¹⁰	2090	46	46	200	55%	2048	21%
11	Uglašen ₉ ¹⁰	2090	48	48	200	55%	2048	12%
12	Uglašen ₁₀ ¹⁰	2090	47	47	200	54%	2048	14%
13	Uglašen ₁₁ ¹⁰	2061	47	47	200	51%	2048	13%
14	Uglašen ₁₂ ¹⁰	2061	48	48	200	51%	2048	11%
15	Uglašen ₁₃ ¹⁰	2053	48	48	200	49%	2048	11%
16	Uglašen ₁₄ ¹⁰	2045	47	47	200	49%	2048	14%
17	Uglašen ₁₅ ¹⁰	2017	48	48	200	46%	2048	11%
18	Uglašen ₁₆ ¹⁰	2010	48	48	200	45%	2048	8%
19	Uglašen ₁₇ ¹⁰	2007	49	49	200	46%	2048	9%
20	Uglašen ₁₈ ¹⁰	2004	48	48	200	45%	2048	10%
21	Uglašen ₁₉ ¹⁰	1986	49	49	200	43%	2048	10%
22	Uglašen ₂₀ ¹⁰	1974	46	46	200	43%	2048	18%
23	Rybka ₁₉₀₀	1949	19	19	1050	36%	2061	14%
24	Začetni ¹⁰	1940	50	50	200	39%	2048	8%
25	Rybka ₁₇₀₀	1781	23	23	1050	18%	2061	9%

(c) 3. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	21	21	1050	78%	2075	14%
2	Uglašen ¹⁰ ₁	2195	48	48	200	62%	2077	8%
3	Rybka2100	2191	19	19	1050	65%	2075	17%
4	Uglašen ¹⁰ ₂	2166	46	46	200	59%	2077	16%
5	Uglašen ¹⁰ ₃	2135	47	47	200	56%	2077	9%
6	Uglašen ¹⁰ ₄	2129	47	47	200	55%	2077	8%
7	Uglašen ¹⁰ ₅	2125	47	47	200	55%	2077	9%
8	Uglašen ¹⁰ ₆	2117	46	46	200	54%	2077	17%
9	Uglašen ¹⁰ ₇	2112	46	46	200	53%	2077	17%
10	Uglašen ¹⁰ ₈	2088	46	46	200	51%	2077	17%
11	Uglašen ¹⁰ ₉	2078	47	47	200	49%	2077	11%
12	Uglašen ¹⁰ ₁₀	2061	46	46	200	48%	2077	17%
13	Uglašen ¹⁰ ₁₁	2059	46	46	200	47%	2077	15%
14	Uglašen ¹⁰ ₁₂	2054	47	47	200	46%	2077	16%
15	Uglašen ¹⁰ ₁₃	2053	46	46	200	46%	2077	18%
16	Uglašen ¹⁰ ₁₄	2047	46	46	200	48%	2077	24%
17	Uglašen ¹⁰ ₁₅	2046	45	45	200	47%	2077	19%
18	Uglašen ¹⁰ ₁₆	2040	46	46	200	45%	2077	13%
19	Uglašen ¹⁰ ₁₇	2034	47	47	200	44%	2077	14%
20	Uglašen ¹⁰ ₁₈	2029	47	47	200	45%	2077	15%
21	Uglašen ¹⁰ ₁₉	2017	49	49	200	43%	2077	11%
22	Rybka1900	2017	19	19	1050	42%	2075	15%
23	Uglašen ¹⁰ ₂₀	2006	48	48	200	42%	2077	14%
24	Začetni ¹⁰	1975	48	48	200	39%	2077	9%
25	Rybka1700	1800	23	23	1050	18%	2075	10%

(d) 4. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	23	23	1050	84%	2010	11%
2	Rybka2100	2159	20	20	1050	68%	2010	10%
3	Uglašen ¹⁰ ₁	2114	49	49	200	57%	2028	12%
4	Uglašen ¹⁰ ₂	2089	49	49	200	55%	2028	15%
5	Uglašen ¹⁰ ₃	2084	49	49	200	55%	2028	11%
6	Uglašen ¹⁰ ₄	2082	50	50	200	55%	2028	6%
7	Uglašen ¹⁰ ₅	2047	49	49	200	50%	2028	14%
8	Uglašen ¹⁰ ₆	2043	50	50	200	50%	2028	5%
9	Uglašen ¹⁰ ₇	2042	49	49	200	51%	2028	10%
10	Uglašen ¹⁰ ₈	2026	48	48	200	48%	2028	17%
11	Uglašen ¹⁰ ₉	2024	50	50	200	48%	2028	9%
12	Uglašen ¹⁰ ₁₀	2020	49	49	200	49%	2028	9%
13	Uglašen ¹⁰ ₁₁	1999	49	49	200	46%	2028	11%
14	Uglašen ¹⁰ ₁₂	1999	48	48	200	45%	2028	16%
15	Uglašen ¹⁰ ₁₃	1998	50	50	200	46%	2028	11%
16	Uglašen ¹⁰ ₁₄	1992	50	50	200	45%	2028	8%
17	Rybka1900	1985	19	19	1050	47%	2010	17%
18	Uglašen ¹⁰ ₁₅	1977	50	50	200	44%	2028	9%
19	Uglašen ¹⁰ ₁₆	1969	48	48	200	43%	2028	24%
20	Uglašen ¹⁰ ₁₇	1969	49	49	200	43%	2028	14%
21	Uglašen ¹⁰ ₁₈	1967	50	50	200	43%	2028	10%
22	Uglašen ¹⁰ ₁₉	1952	49	49	200	42%	2028	12%
23	Uglašen ¹⁰ ₂₀	1915	51	51	200	37%	2028	9%
24	Začetni ¹⁰	1905	54	54	200	37%	2028	4%
25	Rybka1700	1666	26	26	1050	14%	2010	6%

(e) 5. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	23	23	1050	82%	2022	14%
2	Rybka2100	2166	20	20	1050	68%	2022	15%
3	Uglašen ¹⁰ ₁	2132	48	48	200	58%	2044	8%
4	Uglašen ¹⁰ ₂	2105	46	46	200	56%	2044	19%
5	Uglašen ¹⁰ ₃	2091	47	47	200	53%	2044	16%
6	Uglašen ¹⁰ ₄	2083	49	49	200	53%	2044	9%
7	Uglašen ¹⁰ ₅	2071	47	47	200	52%	2044	16%
8	Uglašen ¹⁰ ₆	2070	47	47	200	52%	2044	11%
9	Uglašen ¹⁰ ₇	2058	47	47	200	52%	2044	14%
10	Uglašen ¹⁰ ₈	2048	47	47	200	50%	2044	15%
11	Uglašen ¹⁰ ₉	2043	48	48	200	49%	2044	15%
12	Uglašen ¹⁰ ₁₀	2040	48	48	200	49%	2044	14%
13	Uglašen ¹⁰ ₁₁	2032	47	47	200	49%	2044	16%
14	Uglašen ¹⁰ ₁₂	2021	48	48	200	47%	2044	10%
15	Uglašen ¹⁰ ₁₃	2014	45	45	200	46%	2044	28%
16	Uglašen ¹⁰ ₁₄	2008	48	48	200	45%	2044	11%
17	Uglašen ¹⁰ ₁₅	2008	48	48	200	46%	2044	10%
18	Uglašen ¹⁰ ₁₆	2006	47	47	200	46%	2044	17%
19	Uglašen ¹⁰ ₁₇	2006	48	48	200	44%	2044	14%
20	Uglašen ¹⁰ ₁₈	1987	48	48	200	43%	2044	13%
21	Uglašen ¹⁰ ₁₉	1978	50	50	200	43%	2044	9%
22	Rybka1900	1959	19	19	1050	42%	2022	15%
23	Uglašen ¹⁰ ₂₀	1913	49	49	200	37%	2044	12%
24	Začetni ¹⁰	1760	57	57	200	23%	2044	5%
25	Rybka1700	1753	23	23	1050	19%	2022	8%

(f) 6. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	24	24	1050	84%	2000	10%
2	Rybka2100	2128	20	20	1050	66%	2000	12%
3	Uglašen ¹⁰ ₁	2082	48	48	200	56%	2012	11%
4	Uglašen ¹⁰ ₂	2060	50	50	200	54%	2012	5%
5	Uglašen ¹⁰ ₃	2059	49	49	200	54%	2012	7%
6	Uglašen ¹⁰ ₄	2054	49	49	200	54%	2012	8%
7	Uglašen ¹⁰ ₅	2049	49	49	200	53%	2012	12%
8	Uglašen ¹⁰ ₆	2035	47	47	200	52%	2012	17%
9	Uglašen ¹⁰ ₇	2027	49	49	200	51%	2012	11%
10	Uglašen ¹⁰ ₈	2022	48	48	200	50%	2012	13%
11	Uglašen ¹⁰ ₉	2019	49	49	200	49%	2012	11%
12	Uglašen ¹⁰ ₁₀	2014	50	50	200	50%	2012	7%
13	Uglašen ¹⁰ ₁₁	2014	49	49	200	49%	2012	12%
14	Uglašen ¹⁰ ₁₂	2013	48	48	200	50%	2012	12%
15	Uglašen ¹⁰ ₁₃	2001	48	48	200	48%	2012	10%
16	Uglašen ¹⁰ ₁₄	1995	50	50	200	48%	2012	10%
17	Uglašen ¹⁰ ₁₅	1981	48	48	200	48%	2012	13%
18	Uglašen ¹⁰ ₁₆	1980	50	50	200	46%	2012	9%
19	Uglašen ¹⁰ ₁₇	1969	50	50	200	45%	2012	12%
20	Uglašen ¹⁰ ₁₈	1956	49	49	200	44%	2012	11%
21	Uglašen ¹⁰ ₁₉	1948	51	51	200	43%	2012	9%
22	Rybka1900	1930	19	19	1050	41%	2000	12%
23	Uglašen ¹⁰ ₂₀	1864	52	52	200	35%	2012	6%
24	Začetni ¹⁰	1860	50	50	200	35%	2012	15%
25	Rybka1700	1691	24	24	1050	16%	2000	7%

(g) 7. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	79%	2062	13%
2	Uglašen ¹⁰ ₁	2202	48	48	200	66%	2054	15%
3	Uglašen ¹⁰ ₂	2146	48	48	200	60%	2054	12%
4	Rybka ₂₁₀₀	2141	19	19	1050	61%	2062	16%
5	Uglašen ¹⁰ ₃	2120	47	47	200	57%	2054	13%
6	Uglašen ¹⁰ ₄	2119	48	48	200	56%	2054	10%
7	Uglašen ¹⁰ ₅	2116	47	47	200	56%	2054	13%
8	Uglašen ¹⁰ ₆	2114	47	47	200	56%	2054	9%
9	Uglašen ¹⁰ ₇	2111	46	46	200	56%	2054	16%
10	Uglašen ¹⁰ ₈	2088	47	47	200	53%	2054	13%
11	Uglašen ¹⁰ ₉	2080	48	48	200	52%	2054	8%
12	Uglašen ¹⁰ ₁₀	2067	43	43	200	52%	2054	34%
13	Uglašen ¹⁰ ₁₁	2055	47	47	200	51%	2054	13%
14	Uglašen ¹⁰ ₁₂	2044	47	47	200	49%	2054	11%
15	Uglašen ¹⁰ ₁₃	2043	47	47	200	49%	2054	15%
16	Uglašen ¹⁰ ₁₄	2042	46	46	200	50%	2054	17%
17	Uglašen ¹⁰ ₁₅	2036	47	47	200	49%	2054	14%
18	Uglašen ¹⁰ ₁₆	2030	45	45	200	47%	2054	20%
19	Uglašen ¹⁰ ₁₇	2027	47	47	200	47%	2054	14%
20	Uglašen ¹⁰ ₁₈	2026	48	48	200	47%	2054	7%
21	Uglašen ¹⁰ ₁₉	2001	48	48	200	44%	2054	10%
22	Rybka ₁₉₀₀	1983	19	19	1050	40%	2062	16%
23	Začetni ¹⁰	1926	48	48	200	36%	2054	10%
24	Uglašen ¹⁰ ₂₀	1915	50	50	200	33%	2054	16%
25	Rybka ₁₇₀₀	1787	23	23	1050	19%	2062	9%

(h) 8. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	21	21	1050	78%	2074	14%
2	Rybka ₂₁₀₀	2189	19	19	1050	65%	2074	16%
3	Uglašen ¹⁰ ₁	2152	49	49	200	60%	2042	12%
4	Uglašen ¹⁰ ₂	2150	49	49	200	60%	2042	11%
5	Uglašen ¹⁰ ₃	2135	47	47	200	59%	2042	17%
6	Uglašen ¹⁰ ₄	2119	48	48	200	56%	2042	13%
7	Uglašen ¹⁰ ₅	2118	49	49	200	57%	2042	11%
8	Uglašen ¹⁰ ₆	2110	49	49	200	57%	2042	15%
9	Uglašen ¹⁰ ₇	2102	48	48	200	56%	2042	11%
10	Uglašen ¹⁰ ₈	2099	49	49	200	55%	2042	9%
11	Uglašen ¹⁰ ₉	2093	47	47	200	54%	2042	18%
12	Uglašen ¹⁰ ₁₀	2090	48	48	200	54%	2042	13%
13	Uglašen ¹⁰ ₁₁	2083	49	49	200	53%	2042	12%
14	Uglašen ¹⁰ ₁₂	2075	49	49	200	52%	2042	9%
15	Uglašen ¹⁰ ₁₃	2070	49	49	200	51%	2042	13%
16	Uglašen ¹⁰ ₁₄	2063	49	49	200	51%	2042	13%
17	Uglašen ¹⁰ ₁₅	2055	48	48	200	50%	2042	13%
18	Uglašen ¹⁰ ₁₆	2048	45	45	200	51%	2042	33%
19	Uglašen ¹⁰ ₁₇	2038	49	49	200	47%	2042	12%
20	Uglašen ¹⁰ ₁₈	2007	49	49	200	45%	2042	10%
21	Uglašen ¹⁰ ₁₉	1990	49	49	200	44%	2042	14%
22	Uglašen ¹⁰ ₂₀	1989	48	48	200	44%	2042	16%
23	Rybka ₁₉₀₀	1977	19	19	1050	37%	2074	16%
24	Začetni ¹⁰	1974	49	49	200	42%	2042	10%
25	Rybka ₁₇₀₀	1701	27	27	1050	11%	2074	7%

(i) 9. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	81%	2043	12%
2	Rybka ₂₁₀₀	2184	20	20	1050	68%	2043	13%
3	Uglašen ¹⁰ ₁	2125	48	48	200	57%	2051	10%
4	Uglašen ¹⁰ ₂	2102	49	49	200	56%	2051	6%
5	Uglašen ¹⁰ ₃	2089	48	48	200	53%	2051	12%
6	Uglašen ¹⁰ ₄	2080	46	46	200	52%	2051	18%
7	Uglašen ¹⁰ ₅	2073	48	48	200	51%	2051	14%
8	Uglašen ¹⁰ ₆	2073	48	48	200	52%	2051	11%
9	Uglašen ¹⁰ ₇	2071	47	47	200	52%	2051	14%
10	Uglašen ¹⁰ ₈	2066	48	48	200	51%	2051	11%
11	Uglašen ¹⁰ ₉	2054	48	48	200	50%	2051	11%
12	Uglašen ¹⁰ ₁₀	2049	47	47	200	50%	2051	18%
13	Uglašen ¹⁰ ₁₁	2047	48	48	200	49%	2051	13%
14	Uglašen ¹⁰ ₁₂	2045	47	47	200	49%	2051	13%
15	Uglašen ¹⁰ ₁₃	2035	48	48	200	47%	2051	13%
16	Uglašen ¹⁰ ₁₄	2032	48	48	200	49%	2051	10%
17	Uglašen ¹⁰ ₁₅	2030	48	48	200	47%	2051	10%
18	Uglašen ¹⁰ ₁₆	1994	49	49	200	43%	2051	12%
19	Uglašen ¹⁰ ₁₇	1994	48	48	200	44%	2051	12%
20	Uglašen ¹⁰ ₁₈	1994	48	48	200	44%	2051	13%
21	Uglašen ¹⁰ ₁₉	1983	47	47	200	42%	2051	20%
22	Uglašen ¹⁰ ₂₀	1983	48	48	200	43%	2051	13%
23	Začetni ¹⁰	1979	50	50	200	42%	2051	6%
24	Rybka ₁₉₀₀	1957	19	19	1050	39%	2043	16%
25	Rybka ₁₇₀₀	1763	23	23	1050	18%	2043	8%

(j) 10. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	81%	2040	12%
2	Rybka ₂₁₀₀	2146	19	19	1050	63%	2040	14%
3	Uglašen ¹⁰ ₁	2127	46	46	200	58%	2047	18%
4	Uglašen ¹⁰ ₂	2097	46	46	200	55%	2047	17%
5	Uglašen ¹⁰ ₃	2094	46	46	200	54%	2047	13%
6	Uglašen ¹⁰ ₄	2085	48	48	200	53%	2047	11%
7	Uglašen ¹⁰ ₅	2080	48	48	200	54%	2047	7%
8	Uglašen ¹⁰ ₆	2077	47	47	200	52%	2047	20%
9	Uglašen ¹⁰ ₇	2075	47	47	200	53%	2047	13%
10	Uglašen ¹⁰ ₈	2074	49	49	200	52%	2047	7%
11	Uglašen ¹⁰ ₉	2059	46	46	200	51%	2047	17%
12	Uglašen ¹⁰ ₁₀	2044	46	46	200	49%	2047	17%
13	Uglašen ¹⁰ ₁₁	2043	47	47	200	49%	2047	15%
14	Uglašen ¹⁰ ₁₂	2041	46	46	200	48%	2047	18%
15	Uglašen ¹⁰ ₁₃	2040	48	48	200	48%	2047	13%
16	Uglašen ¹⁰ ₁₄	2025	46	46	200	50%	2047	21%
17	Uglašen ¹⁰ ₁₅	2017	47	47	200	46%	2047	16%
18	Uglašen ¹⁰ ₁₆	2017	46	46	200	46%	2047	15%
19	Rybka ₁₉₀₀	1997	19	19	1050	44%	2040	19%
20	Uglašen ¹⁰ ₁₇	1994	47	47	200	44%	2047	15%
21	Uglašen ¹⁰ ₁₈	1994	50	50	200	44%	2047	5%
22	Uglašen ¹⁰ ₁₉	1971	49	49	200	41%	2047	8%
23	Uglašen ¹⁰ ₂₀	1953	50	50	200	39%	2047	9%
24	Začetni ¹⁰	1927	50	50	200	38%	2047	8%
25	Rybka ₁₇₀₀	1746	24	24	1050	17%	2040	8%

Tabela 5.2: Rezultati uglasovanja na ozkih intervalih (*igre* = 5)

(a) 1. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	80%	2042	11%
2	Rybka ₂₁₀₀	2135	19	19	1050	62%	2042	15%
3	Uglašen ₁ ⁵	2127	50	50	200	60%	2023	9%
4	Uglašen ₂ ⁵	2115	48	48	200	60%	2023	15%
5	Uglašen ₃ ⁵	2096	47	47	200	57%	2022	16%
6	Uglašen ₄ ⁵	2089	48	48	200	56%	2023	14%
7	Uglašen ₅ ⁵	2088	49	49	200	56%	2023	9%
8	Uglašen ₆ ⁵	2086	46	46	200	56%	2023	17%
9	Uglašen ₇ ⁵	2073	50	50	200	55%	2023	7%
10	Uglašen ₈ ⁵	2067	49	49	200	54%	2023	7%
11	Uglašen ₉ ⁵	2057	47	47	200	54%	2023	14%
12	Uglašen ₁₀ ⁵	2056	48	48	200	55%	2023	15%
13	Uglašen ₁₁ ⁵	2041	47	47	200	52%	2023	20%
14	Uglašen ₁₂ ⁵	2038	47	47	200	53%	2023	16%
15	Uglašen ₁₃ ⁵	2035	48	48	200	51%	2023	11%
16	Uglašen ₁₄ ⁵	2028	46	46	200	51%	2023	24%
17	Uglašen ₁₅ ⁵	2024	48	48	200	50%	2023	17%
18	Uglašen ₁₆ ⁵	2023	47	47	200	50%	2023	13%
19	Uglašen ₁₇ ⁵	2019	49	49	200	49%	2023	8%
20	Uglašen ₁₈ ⁵	2006	49	49	200	48%	2023	7%
21	Uglašen ₁₉ ⁵	1969	49	49	200	43%	2023	14%
22	Uglašen ₂₀ ⁵	1938	50	50	200	40%	2023	6%
23	Rybka ₁₉₀₀	1930	19	19	1050	36%	2042	17%
24	Začetni ⁵	1907	48	48	200	38%	2023	16%
25	Rybka ₁₇₀₀	1727	25	25	1050	15%	2042	8%

(b) 2. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	80%	2051	11%
2	Rybka ₂₁₀₀	2174	20	20	1050	65%	2051	11%
3	Uglašen ₁ ⁵	2131	47	47	200	57%	2065	14%
4	Uglašen ₂ ⁵	2120	48	48	200	56%	2065	9%
5	Uglašen ₃ ⁵	2116	47	47	200	56%	2065	13%
6	Uglašen ₄ ⁵	2104	48	48	200	55%	2065	12%
7	Uglašen ₅ ⁵	2085	48	48	200	52%	2065	8%
8	Uglašen ₆ ⁵	2079	46	46	200	51%	2065	16%
9	Uglašen ₇ ⁵	2067	46	46	200	50%	2065	15%
10	Uglašen ₈ ⁵	2059	46	46	200	50%	2065	12%
11	Uglašen ₉ ⁵	2059	47	47	200	49%	2065	8%
12	Uglašen ₁₀ ⁵	2057	48	48	200	50%	2065	11%
13	Uglašen ₁₁ ⁵	2051	47	47	200	47%	2065	10%
14	Uglašen ₁₂ ⁵	2040	46	46	200	48%	2065	17%
15	Uglašen ₁₃ ⁵	2039	49	49	200	46%	2065	7%
16	Uglašen ₁₄ ⁵	2038	45	45	200	46%	2065	22%
17	Uglašen ₁₅ ⁵	2037	47	47	200	48%	2065	11%
18	Uglašen ₁₆ ⁵	2026	48	48	200	45%	2065	9%
19	Uglašen ₁₇ ⁵	2025	48	48	200	46%	2065	11%
20	Uglašen ₁₈ ⁵	2017	47	47	200	46%	2065	13%
21	Uglašen ₁₉ ⁵	1981	47	47	200	41%	2065	18%
22	Rybka ₁₉₀₀	1979	19	19	1050	41%	2051	16%
23	Uglašen ₂₀ ⁵	1969	47	47	200	40%	2065	18%
24	Začetni ⁵	1966	48	48	200	39%	2065	10%
25	Rybka ₁₇₀₀	1807	22	22	1050	21%	2051	10%

(c) 3. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	80%	2039	11%
2	Rybka ₂₁₀₀	2138	19	19	1050	63%	2039	15%
3	Uglašen ₁ ⁵	2116	48	48	200	59%	2034	10%
4	Uglašen ₂ ⁵	2113	48	48	200	58%	2034	14%
5	Uglašen ₃ ⁵	2110	48	48	200	58%	2034	10%
6	Uglašen ₄ ⁵	2104	48	48	200	57%	2034	10%
7	Uglašen ₅ ⁵	2101	47	47	200	57%	2034	17%
8	Uglašen ₆ ⁵	2100	49	49	200	56%	2034	8%
9	Uglašen ₇ ⁵	2098	48	48	200	56%	2034	13%
10	Uglašen ₈ ⁵	2089	49	49	200	56%	2034	7%
11	Uglašen ₉ ⁵	2082	46	46	200	55%	2034	19%
12	Uglašen ₁₀ ⁵	2059	48	48	200	53%	2034	12%
13	Uglašen ₁₁ ⁵	2044	49	49	200	49%	2034	11%
14	Uglašen ₁₂ ⁵	2036	48	48	200	50%	2034	10%
15	Uglašen ₁₃ ⁵	2030	48	48	200	50%	2034	15%
16	Uglašen ₁₄ ⁵	2027	47	47	200	50%	2034	14%
17	Uglašen ₁₅ ⁵	1994	47	47	200	45%	2034	14%
18	Uglašen ₁₆ ⁵	1987	48	48	200	44%	2034	12%
19	Uglašen ₁₇ ⁵	1978	49	49	200	45%	2034	11%
20	Uglašen ₁₈ ⁵	1977	48	48	200	45%	2034	12%
21	Uglašen ₁₉ ⁵	1957	47	47	200	42%	2034	15%
22	Rybka ₁₉₀₀	1946	19	19	1050	38%	2039	14%
23	Uglašen ₂₀ ⁵	1941	47	47	200	40%	2034	17%
24	Začetni ⁵	1872	51	51	200	33%	2034	8%
25	Rybka ₁₇₀₀	1750	24	24	1050	18%	2039	8%

(d) 4. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	21	21	1050	77%	2068	13%
2	Uglašen ₁ ⁵	2177	48	48	200	62%	2062	11%
3	Rybka ₂₁₀₀	2161	19	19	1050	62%	2068	16%
4	Uglašen ₂ ⁵	2143	47	47	200	59%	2062	19%
5	Uglašen ₃ ⁵	2124	48	48	200	57%	2062	10%
6	Uglašen ₄ ⁵	2122	47	47	200	57%	2062	13%
7	Uglašen ₅ ⁵	2108	46	46	200	55%	2062	14%
8	Uglašen ₆ ⁵	2106	47	47	200	54%	2061	13%
9	Uglašen ₇ ⁵	2101	46	46	200	54%	2062	15%
10	Uglašen ₈ ⁵	2100	47	47	200	53%	2062	11%
11	Uglašen ₉ ⁵	2099	47	47	200	53%	2062	12%
12	Uglašen ₁₀ ⁵	2074	46	46	200	52%	2062	14%
13	Uglašen ₁₁ ⁵	2073	47	47	200	52%	2062	6%
14	Uglašen ₁₂ ⁵	2071	47	47	200	51%	2062	8%
15	Uglašen ₁₃ ⁵	2068	46	46	200	51%	2062	15%
16	Uglašen ₁₄ ⁵	2065	46	46	200	49%	2062	14%
17	Uglašen ₁₅ ⁵	2061	46	46	200	49%	2062	13%
18	Uglašen ₁₆ ⁵	2061	47	47	200	50%	2062	11%
19	Uglašen ₁₇ ⁵	2053	48	48	200	48%	2062	11%
20	Uglašen ₁₈ ⁵	2049	46	46	200	48%	2062	15%
21	Uglašen ₁₉ ⁵	2036	46	46	200	47%	2062	17%
22	Uglašen ₂₀ ⁵	1991	49	49	200	42%	2062	13%
23	Rybka ₁₉₀₀	1970	20	20	1050	38%	2068	12%
24	Rybka ₁₇₀₀	1819	23	23	1050	21%	2068	8%
25	Začetni ⁵	1745	58	58	200	20%	2062	5%

(e) 5. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	80%	2044	11%
2	Rybka ₂₁₀₀	2170	20	20	1050	66%	2044	13%
3	Uglašen ₁ ⁵	2116	47	47	200	59%	2050	19%
4	Uglašen ₂ ⁵	2116	49	49	200	56%	2050	6%
5	Uglašen ₃ ⁵	2110	48	48	200	56%	2050	12%
6	Uglašen ₄ ⁵	2106	48	48	200	55%	2050	12%
7	Uglašen ₅ ⁵	2105	46	46	200	55%	2050	19%
8	Uglašen ₆ ⁵	2077	48	48	200	53%	2050	9%
9	Uglašen ₇ ⁵	2074	48	48	200	52%	2050	15%
10	Uglašen ₈ ⁵	2066	48	48	200	51%	2050	14%
11	Uglašen ₉ ⁵	2057	48	48	200	50%	2050	9%
12	Uglašen ₁₀ ⁵	2048	48	48	200	49%	2050	12%
13	Uglašen ₁₁ ⁵	2043	49	49	200	49%	2050	7%
14	Uglašen ₁₂ ⁵	2035	47	47	200	48%	2050	13%
15	Uglašen ₁₃ ⁵	2035	49	49	200	48%	2050	10%
16	Uglašen ₁₄ ⁵	2028	50	50	200	46%	2050	6%
17	Uglašen ₁₅ ⁵	2009	49	49	200	45%	2050	7%
18	Uglašen ₁₆ ⁵	2009	48	48	200	45%	2050	10%
19	Uglašen ₁₇ ⁵	1992	48	48	200	43%	2050	11%
20	Rybka ₁₉₀₀	1984	19	19	1050	42%	2044	13%
21	Uglašen ₁₈ ⁵	1983	48	48	200	42%	2050	10%
22	Uglašen ₁₉ ⁵	1979	47	47	200	43%	2050	18%
23	Uglašen ₂₀ ⁵	1974	49	49	200	41%	2050	13%
24	Začetni ⁵	1963	50	50	200	40%	2050	9%
25	Rybka ₁₇₀₀	1746	24	24	1050	17%	2044	8%

(f) 6. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	23	23	1050	82%	2000	12%
3	Rybka ₂₁₀₀	2117	19	19	1050	63%	2000	15%
2	Uglašen ₁ ⁵	2109	48	48	200	60%	2005	14%
4	Uglašen ₂ ⁵	2067	48	48	200	56%	2005	11%
5	Uglašen ₃ ⁵	2049	48	48	200	54%	2005	14%
6	Uglašen ₄ ⁵	2048	48	48	200	54%	2005	12%
7	Uglašen ₅ ⁵	2047	47	47	200	55%	2005	16%
8	Uglašen ₆ ⁵	2039	46	46	200	55%	2005	21%
9	Uglašen ₇ ⁵	2038	47	47	200	53%	2005	17%
10	Uglašen ₈ ⁵	2033	47	47	200	52%	2005	13%
11	Uglašen ₉ ⁵	2021	47	47	200	51%	2005	14%
12	Uglašen ₁₀ ⁵	2020	48	48	200	52%	2005	10%
13	Uglašen ₁₁ ⁵	2015	48	48	200	51%	2005	10%
14	Uglašen ₁₂ ⁵	2009	49	49	200	50%	2005	12%
15	Uglašen ₁₃ ⁵	1997	47	47	200	49%	2005	17%
16	Uglašen ₁₄ ⁵	1993	48	48	200	50%	2005	12%
17	Uglašen ₁₅ ⁵	1987	47	47	200	48%	2005	13%
18	Uglašen ₁₆ ⁵	1973	49	49	200	47%	2005	8%
19	Uglašen ₁₇ ⁵	1970	47	47	200	46%	2005	14%
20	Uglašen ₁₈ ⁵	1958	48	48	200	45%	2005	10%
21	Rybka ₁₉₀₀	1925	19	19	1050	40%	2000	16%
22	Uglašen ₁₉ ⁵	1924	49	49	200	40%	2005	14%
23	Uglašen ₂₀ ⁵	1919	49	49	200	41%	2005	13%
24	Začetni ⁵	1755	54	54	200	26%	2005	9%
25	Rybka ₁₇₀₀	1714	24	24	1050	18%	2000	9%

(g) 7. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	23	23	1050	82%	2027	12%
2	Uglašen ₁ ⁵	2170	48	48	200	64%	2032	14%
3	Rybka ₂₁₀₀	2134	19	19	1050	63%	2027	15%
4	Uglašen ₂ ⁵	2094	48	48	200	56%	2032	11%
5	Uglašen ₃ ⁵	2076	48	48	200	54%	2032	14%
6	Uglašen ₄ ⁵	2075	48	48	200	54%	2032	12%
7	Uglašen ₅ ⁵	2074	47	47	200	55%	2032	16%
8	Uglašen ₆ ⁵	2066	46	46	200	55%	2032	21%
9	Uglašen ₇ ⁵	2065	47	47	200	53%	2032	17%
10	Uglašen ₈ ⁵	2060	47	47	200	52%	2032	13%
11	Uglašen ₉ ⁵	2048	47	47	200	51%	2032	14%
12	Uglašen ₁₀ ⁵	2047	48	48	200	52%	2032	10%
13	Uglašen ₁₁ ⁵	2042	48	48	200	51%	2032	10%
14	Uglašen ₁₂ ⁵	2036	49	49	200	50%	2032	12%
15	Uglašen ₁₃ ⁵	2024	47	47	200	49%	2032	17%
16	Uglašen ₁₄ ⁵	2020	48	48	200	50%	2032	12%
17	Uglašen ₁₅ ⁵	2014	47	47	200	48%	2032	13%
18	Uglašen ₁₆ ⁵	2000	49	49	200	47%	2032	8%
19	Uglašen ₁₇ ⁵	1997	47	47	200	46%	2032	14%
20	Uglašen ₁₈ ⁵	1985	48	48	200	45%	2032	10%
21	Rybka ₁₉₀₀	1952	19	19	1050	40%	2027	16%
22	Uglašen ₁₉ ⁵	1951	49	49	200	40%	2032	14%
23	Uglašen ₂₀ ⁵	1946	49	49	200	41%	2032	13%
24	Začetni ⁵	1782	54	54	200	26%	2032	9%
25	Rybka ₁₇₀₀	1741	24	24	1050	18%	2027	9%

(h) 8. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	79%	2060	12%
2	Uglašen ₁ ⁵	2147	49	49	200	63%	2029	15%
3	Rybka ₂₁₀₀	2135	19	19	1050	60%	2060	17%
4	Uglašen ₂ ⁵	2134	47	47	200	61%	2029	19%
5	Uglašen ₃ ⁵	2101	48	48	200	57%	2029	10%
6	Uglašen ₄ ⁵	2101	48	48	200	59%	2029	16%
7	Uglašen ₅ ⁵	2093	47	47	200	57%	2029	17%
8	Uglašen ₆ ⁵	2085	49	49	200	56%	2029	9%
9	Uglašen ₇ ⁵	2082	47	47	200	56%	2029	12%
10	Uglašen ₈ ⁵	2071	48	48	200	54%	2027	14%
11	Uglašen ₉ ⁵	2068	48	48	200	54%	2029	14%
12	Uglašen ₁₀ ⁵	2065	47	47	200	54%	2029	15%
13	Uglašen ₁₁ ⁵	2064	49	49	200	53%	2029	9%
14	Uglašen ₁₂ ⁵	2057	48	48	200	53%	2029	11%
15	Uglašen ₁₃ ⁵	2057	48	48	200	52%	2029	10%
16	Uglašen ₁₄ ⁵	2056	48	48	200	53%	2029	13%
17	Uglašen ₁₅ ⁵	2054	47	47	200	52%	2029	15%
18	Uglašen ₁₆ ⁵	2049	47	47	200	52%	2029	11%
19	Uglašen ₁₇ ⁵	2044	47	47	200	53%	2029	19%
20	Uglašen ₁₈ ⁵	2014	49	49	200	48%	2029	11%
21	Uglašen ₁₉ ⁵	2014	47	47	200	48%	2029	17%
22	Začetni ⁵	1961	49	49	200	43%	2029	9%
23	Rybka ₁₉₀₀	1952	19	19	1050	36%	2060	15%
24	Uglašen ₂₀ ⁵	1946	50	50	200	40%	2029	11%
25	Rybka ₁₇₀₀	1727	25	25	1050	13%	2060	9%

(i) 9. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	81%	2048	13%
2	Rybka ₂₁₀₀	2146	19	19	1050	63%	2048	16%
3	Uglašen ₁ ⁵	2111	46	46	200	57%	2045	19%
4	Uglašen ₂ ⁵	2102	47	47	200	56%	2045	13%
5	Uglašen ₃ ⁵	2084	47	47	200	53%	2045	10%
6	Uglašen ₄ ⁵	2079	48	48	200	53%	2045	14%
7	Uglašen ₅ ⁵	2070	47	47	200	52%	2045	10%
8	Uglašen ₆ ⁵	2067	47	47	200	52%	2045	11%
9	Uglašen ₇ ⁵	2062	46	46	200	53%	2045	21%
10	Uglašen ₈ ⁵	2057	48	48	200	51%	2045	10%
11	Uglašen ₉ ⁵	2045	47	47	200	50%	2045	16%
12	Uglašen ₁₀ ⁵	2044	47	47	200	50%	2045	11%
13	Začetni ⁵	2043	47	47	200	50%	2045	13%
14	Uglašen ₁₁ ⁵	2043	48	48	200	50%	2045	9%
15	Uglašen ₁₂ ⁵	2041	47	47	200	51%	2045	18%
16	Uglašen ₁₃ ⁵	2037	47	47	200	49%	2045	13%
17	Uglašen ₁₄ ⁵	2034	46	46	200	48%	2045	20%
18	Uglašen ₁₅ ⁵	2032	47	47	200	50%	2045	16%
19	Uglašen ₁₆ ⁵	2031	47	47	200	48%	2045	14%
20	Uglašen ₁₇ ⁵	2025	47	47	200	47%	2045	19%
21	Uglašen ₁₈ ⁵	2025	47	47	200	47%	2045	12%
22	Uglašen ₁₉ ⁵	2011	46	46	200	46%	2045	19%
23	Uglašen ₂₀ ⁵	1976	50	50	200	42%	2045	8%
24	Rybka ₁₉₀₀	1965	19	19	1050	39%	2048	16%
25	Rybka ₁₇₀₀	1772	23	23	1050	18%	2048	10%

(j) 10. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	23	23	1050	83%	2013	11%
2	Rybka ₂₁₀₀	2150	20	20	1050	67%	2013	12%
3	Uglašen ₁ ⁵	2124	47	47	200	57%	2057	12%
4	Uglašen ₂ ⁵	2092	47	47	200	53%	2057	13%
5	Uglašen ₃ ⁵	2086	47	47	200	54%	2057	14%
6	Uglašen ₄ ⁵	2042	46	46	200	49%	2057	23%
7	Uglašen ₅ ⁵	2037	48	48	200	47%	2057	12%
8	Uglašen ₆ ⁵	2029	48	48	200	45%	2057	10%
9	Uglašen ₇ ⁵	2024	48	48	200	46%	2057	7%
10	Uglašen ₈ ⁵	2020	48	48	200	46%	2057	8%
11	Uglašen ₉ ⁵	2018	47	47	200	46%	2057	12%
12	Uglašen ₁₀ ⁵	2015	48	48	200	45%	2057	11%
13	Uglašen ₁₁ ⁵	2013	47	47	200	44%	2057	12%
14	Uglašen ₁₂ ⁵	2013	48	48	200	45%	2057	9%
15	Uglašen ₁₃ ⁵	2013	48	48	200	46%	2057	9%
16	Uglašen ₁₄ ⁵	2011	47	47	200	45%	2057	15%
17	Uglašen ₁₅ ⁵	2009	46	46	200	45%	2057	19%
18	Rybka ₁₉₀₀	2007	19	19	1050	49%	2013	15%
19	Uglašen ₁₆ ⁵	2004	47	47	200	43%	2057	16%
20	Uglašen ₁₇ ⁵	1980	48	48	200	41%	2057	11%
21	Uglašen ₁₈ ⁵	1977	50	50	200	40%	2057	10%
22	Uglašen ₁₉ ⁵	1951	48	48	200	38%	2057	13%
23	Uglašen ₂₀ ⁵	1942	49	49	200	37%	2057	15%
24	Začetni ⁵	1874	50	50	200	32%	2057	11%
25	Rybka ₁₇₀₀	1771	22	22	1050	21%	2013	11%

Tabela 5.3: Rezultati uglaševanja na ozkih intervalih ($igre = 2$)

(a) 1. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	25	25	1050	84%	1985	6%
2	Rybka ₂₁₀₀	2128	20	20	1050	67%	1985	10%
3	Uglašen ₁ ²	2073	48	48	200	54%	2043	10%
4	Uglašen ₂ ²	2072	47	47	200	53%	2043	8%
5	Uglašen ₃ ²	2051	48	48	200	51%	2043	11%
6	Uglašen ₄ ²	2030	47	47	200	49%	2043	9%
7	Uglašen ₅ ²	2023	47	47	200	48%	2043	11%
8	Uglašen ₆ ²	2010	48	48	200	46%	2043	8%
9	Uglašen ₇ ²	1992	48	48	200	44%	2043	10%
10	Uglašen ₈ ²	1991	46	46	200	45%	2043	14%
11	Uglašen ₉ ²	1986	46	46	200	45%	2043	17%
12	Uglašen ₁₀ ²	1984	47	47	200	44%	2043	14%
13	Uglašen ₁₁ ²	1978	47	47	200	44%	2043	16%
14	Uglašen ₁₂ ²	1977	46	46	200	43%	2043	21%
15	Uglašen ₁₃ ²	1973	47	47	200	43%	2043	9%
16	Uglašen ₁₄ ²	1967	47	47	200	42%	2043	11%
17	Uglašen ₁₅ ²	1963	48	48	200	42%	2043	13%
18	Uglašen ₁₆ ²	1959	48	48	200	42%	2043	9%
19	Rybka ₁₉₀₀	1957	19	19	1050	46%	1985	15%
20	Uglašen ₁₇ ²	1957	49	49	200	41%	2043	9%
21	Uglašen ₁₈ ²	1956	47	47	200	41%	2043	17%
22	Uglašen ₁₉ ²	1939	51	51	200	39%	2043	6%
23	Začetni ²	1925	49	49	200	37%	2043	9%
24	Uglašen ₂₀ ²	1873	51	51	200	33%	2043	5%
25	Rybka ₁₇₀₀	1786	21	21	1050	26%	1985	13%

(b) 2. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	22	22	1050	80%	2046	11%
2	Rybka ₂₁₀₀	2199	20	20	1050	68%	2046	10%
3	Uglašen ₁ ²	2130	49	49	200	60%	2049	13%
4	Uglašen ₂ ²	2127	48	48	200	58%	2050	13%
5	Uglašen ₃ ²	2117	48	48	200	56%	2050	11%
6	Uglašen ₄ ²	2097	47	47	200	57%	2050	20%
7	Uglašen ₅ ²	2093	48	48	200	55%	2050	14%
8	Uglašen ₆ ²	2092	48	48	200	54%	2050	11%
9	Uglašen ₇ ²	2084	48	48	200	52%	2050	11%
10	Uglašen ₈ ²	2063	47	47	200	51%	2050	8%
11	Uglašen ₉ ²	2057	48	48	200	50%	2050	11%
12	Uglašen ₁₀ ²	2056	49	49	200	50%	2050	12%
13	Uglašen ₁₁ ²	2051	48	48	200	49%	2050	11%
14	Uglašen ₁₂ ²	2049	47	47	200	50%	2050	10%
15	Uglašen ₁₃ ²	2034	49	49	200	47%	2050	15%
16	Uglašen ₁₄ ²	2023	49	49	200	46%	2050	13%
17	Uglašen ₁₅ ²	2021	48	48	200	47%	2050	9%
18	Uglašen ₁₆ ²	2019	49	49	200	46%	2050	9%
19	Uglašen ₁₇ ²	2018	50	50	200	47%	2050	6%
20	Uglašen ₁₈ ²	2007	48	48	200	45%	2050	10%
21	Uglašen ₁₉ ²	1993	48	48	200	45%	2050	15%
22	Uglašen ₂₀ ²	1988	50	50	200	44%	2050	10%
23	Rybka ₁₉₀₀	1953	19	19	1050	38%	2046	14%
24	Začetni ²	1854	53	53	200	30%	2050	6%
25	Rybka ₁₇₀₀	1751	24	24	1050	17%	2046	9%

(c) 3. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	24	24	1050	85%	1989	9%
2	Rybka ₂₁₀₀	2154	20	20	1050	70%	1989	13%
3	Uglašen ₁ ²	2085	48	48	200	53%	2044	8%
4	Uglašen ₂ ²	2084	48	48	200	53%	2044	11%
5	Uglašen ₃ ²	2059	48	48	200	50%	2044	8%
6	Uglašen ₄ ²	2050	47	47	200	51%	2044	13%
7	Uglašen ₅ ²	2045	47	47	200	50%	2044	13%
8	Uglašen ₆ ²	2029	48	48	200	48%	2044	8%
9	Uglašen ₇ ²	2027	47	47	200	47%	2044	17%
10	Uglašen ₈ ²	2011	48	48	200	46%	2044	11%
11	Uglašen ₉ ²	2010	48	48	200	46%	2044	9%
12	Uglašen ₁₀ ²	2003	48	48	200	45%	2044	13%
13	Uglašen ₁₁ ²	2001	49	49	200	45%	2044	13%
14	Uglašen ₁₂ ²	1997	47	47	200	45%	2044	13%
15	Uglašen ₁₃ ²	1987	49	49	200	44%	2044	9%
16	Rybka ₁₉₀₀	1982	19	19	1050	49%	1989	12%
17	Uglašen ₁₄ ²	1973	49	49	200	42%	2044	10%
18	Uglašen ₁₅ ²	1964	48	48	200	40%	2044	15%
19	Uglašen ₁₆ ²	1961	49	49	200	40%	2044	9%
20	Uglašen ₁₇ ²	1936	48	48	200	39%	2044	16%
21	Začetni ²	1909	50	50	200	36%	2044	9%
22	Uglašen ₁₈ ²	1900	52	52	200	35%	2044	8%
23	Uglašen ₁₉ ²	1877	51	51	200	32%	2044	14%
24	Uglašen ₂₀ ²	1864	52	52	200	31%	2044	8%
25	Rybka ₁₇₀₀	1741	22	22	1050	21%	1989	9%

(d) 4. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	24	24	1050	84%	2003	11%
2	Rybka ₂₁₀₀	2168	20	20	1050	70%	2003	11%
3	Uglašen ₁ ²	2066	49	49	200	53%	2030	12%
4	Uglašen ₂ ²	2058	47	47	200	52%	2030	14%
5	Uglašen ₃ ²	2057	49	49	200	53%	2030	14%
6	Uglašen ₄ ²	2056	49	49	200	52%	2030	12%
7	Uglašen ₅ ²	2053	48	48	200	51%	2030	11%
8	Uglašen ₆ ²	2053	49	49	200	52%	2030	9%
9	Uglašen ₇ ²	2050	49	49	200	52%	2030	10%
10	Uglašen ₈ ²	2041	49	49	200	50%	2030	11%
11	Uglašen ₉ ²	2032	48	48	200	50%	2030	11%
12	Uglašen ₁₀ ²	2031	48	48	200	49%	2030	15%
13	Uglašen ₁₁ ²	2031	49	49	200	49%	2030	8%
14	Uglašen ₁₂ ²	2002	49	49	200	46%	2030	8%
15	Uglašen ₁₃ ²	1999	47	47	200	46%	2030	15%
16	Uglašen ₁₄ ²	1989	48	48	200	45%	2030	11%
17	Uglašen ₁₅ ²	1986	50	50	200	45%	2030	11%
18	Uglašen ₁₆ ²	1967	51	51	200	43%	2030	10%
19	Uglašen ₁₇ ²	1957	49	49	200	42%	2030	9%
20	Rybka ₁₉₀₀	1938	19	19	1050	41%	2003	15%
21	Uglašen ₁₈ ²	1926	50	50	200	40%	2030	12%
22	Uglašen ₁₉ ²	1921	51	51	200	38%	2030	6%
23	Uglašen ₂₀ ²	1918	50	50	200	38%	2030	13%
24	Začetni ²	1862	53	53	200	33%	2030	6%
25	Rybka ₁₇₀₀	1713	24	24	1050	18%	2003	5%

(e) 5. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	23	23	1050	83%	2019	11%
2	Rybka2100	2175	20	20	1050	70%	2019	16%
3	Uglašen ₁ ²	2079	49	49	200	54%	2030	15%
4	Uglašen ₂ ²	2074	48	48	200	54%	2030	11%
5	Uglašen ₃ ²	2073	50	50	200	54%	2030	9%
6	Uglašen ₄ ²	2070	47	47	200	52%	2030	18%
7	Uglašen ₅ ²	2068	48	48	200	53%	2030	11%
8	Uglašen ₆ ²	2062	48	48	200	53%	2030	13%
9	Uglašen ₇ ²	2059	48	48	200	52%	2030	15%
10	Uglašen ₈ ²	2053	49	49	200	52%	2030	6%
11	Uglašen ₉ ²	2039	49	49	200	50%	2030	12%
12	Uglašen ₁₀ ²	2014	49	49	200	48%	2030	11%
13	Uglašen ₁₁ ²	2011	49	49	200	47%	2030	9%
14	Uglašen ₁₂ ²	2005	49	49	200	46%	2030	13%
15	Uglašen ₁₃ ²	2004	48	48	200	45%	2030	17%
16	Uglašen ₁₄ ²	1999	48	48	200	45%	2030	15%
17	Uglašen ₁₅ ²	1997	48	48	200	46%	2030	14%
18	Uglašen ₁₆ ²	1996	49	49	200	46%	2030	10%
19	Uglašen ₁₇ ²	1993	47	47	200	46%	2030	15%
20	Uglašen ₁₈ ²	1983	49	49	200	46%	2030	13%
21	Uglašen ₁₉ ²	1964	50	50	200	43%	2030	9%
22	Uglašen ₂₀ ²	1949	49	49	200	42%	2030	12%
23	Rybka1900	1923	19	19	1050	38%	2019	13%
24	Začetni ²	1914	50	50	200	38%	2030	6%
25	Rybka1700	1723	24	24	1050	17%	2019	7%

(f) 6. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	22	22	1050	80%	2050	13%
2	Rybka2100	2161	19	19	1050	64%	2050	15%
3	Uglašen ₁ ²	2139	48	48	200	60%	2045	14%
4	Uglašen ₂ ²	2132	48	48	200	59%	2045	10%
5	Uglašen ₃ ²	2124	47	47	200	59%	2045	18%
6	Uglašen ₄ ²	2104	48	48	200	55%	2045	11%
7	Uglašen ₅ ²	2102	48	48	200	55%	2045	17%
8	Uglašen ₆ ²	2099	48	48	200	54%	2045	8%
9	Uglašen ₇ ²	2084	48	48	200	53%	2045	12%
10	Uglašen ₈ ²	2066	48	48	200	51%	2045	9%
11	Uglašen ₉ ²	2057	49	49	200	50%	2045	8%
12	Uglašen ₁₀ ²	2048	48	48	200	50%	2045	13%
13	Uglašen ₁₁ ²	2047	47	47	200	51%	2045	19%
14	Uglašen ₁₂ ²	2043	47	47	200	50%	2045	14%
15	Uglašen ₁₃ ²	2028	48	48	200	47%	2045	12%
16	Uglašen ₁₄ ²	2026	48	48	200	48%	2045	13%
17	Uglašen ₁₅ ²	2017	48	48	200	46%	2045	16%
18	Uglašen ₁₆ ²	2017	47	47	200	46%	2045	15%
19	Uglašen ₁₇ ²	2011	47	47	200	46%	2045	13%
20	Uglašen ₁₈ ²	2001	48	48	200	45%	2045	16%
21	Uglašen ₁₉ ²	1991	48	48	200	43%	2045	11%
22	Rybka1900	1985	19	19	1050	42%	2050	14%
23	Začetni ²	1983	47	47	200	42%	2045	22%
24	Uglašen ₂₀ ²	1924	49	49	200	38%	2045	14%
25	Rybka1700	1735	24	24	1050	14%	2050	12%

(g) 7. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	23	23	1050	83%	2021	11%
2	Rybka2100	2174	20	20	1050	69%	2021	13%
3	Uglašen ₁ ²	2101	48	48	200	55%	2052	8%
4	Uglašen ₂ ²	2085	45	45	200	54%	2052	23%
5	Uglašen ₃ ²	2077	49	49	200	51%	2052	9%
6	Uglašen ₄ ²	2053	47	47	200	50%	2052	10%
7	Uglašen ₅ ²	2050	47	47	200	50%	2052	13%
8	Uglašen ₆ ²	2050	48	48	200	49%	2052	8%
9	Uglašen ₇ ²	2048	47	47	200	49%	2052	17%
10	Uglašen ₈ ²	2045	49	49	200	49%	2052	9%
11	Uglašen ₉ ²	2038	49	49	200	47%	2052	11%
12	Uglašen ₁₀ ²	2036	49	49	200	48%	2052	9%
13	Uglašen ₁₁ ²	2035	48	48	200	48%	2052	10%
14	Uglašen ₁₂ ²	2012	47	47	200	46%	2052	14%
15	Uglašen ₁₃ ²	2011	47	47	200	44%	2052	16%
16	Uglašen ₁₄ ²	2010	47	47	200	46%	2052	14%
17	Uglašen ₁₅ ²	2007	48	48	200	45%	2052	12%
18	Uglašen ₁₆ ²	1996	49	49	200	44%	2052	11%
19	Uglašen ₁₇ ²	1987	47	47	200	43%	2052	15%
20	Uglašen ₁₈ ²	1975	48	48	200	41%	2052	12%
21	Rybka1900	1971	19	19	1050	43%	2021	17%
22	Uglašen ₁₉ ²	1955	49	49	200	39%	2052	10%
23	Uglašen ₂₀ ²	1937	49	49	200	38%	2052	10%
24	Začetni ²	1933	49	49	200	37%	2052	15%
25	Rybka1700	1762	23	23	1050	20%	2021	7%

(h) 8. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	24	24	1050	85%	1992	10%
2	Rybka2100	2113	20	20	1050	65%	1992	13%
3	Uglašen ₁ ²	2089	48	48	200	58%	2015	12%
4	Uglašen ₂ ²	2061	48	48	200	54%	2015	17%
5	Uglašen ₃ ²	2059	50	50	200	54%	2015	8%
6	Uglašen ₄ ²	2052	49	49	200	54%	2015	8%
7	Uglašen ₅ ²	2049	49	49	200	52%	2015	12%
8	Uglašen ₆ ²	2028	49	49	200	51%	2015	7%
9	Uglašen ₇ ²	2012	49	49	200	49%	2015	7%
10	Uglašen ₈ ²	2008	49	49	200	49%	2015	11%
11	Uglašen ₉ ²	2006	47	47	200	50%	2015	21%
12	Uglašen ₁₀ ²	2005	49	49	200	49%	2015	5%
13	Uglašen ₁₁ ²	1995	48	48	200	48%	2015	8%
14	Uglašen ₁₂ ²	1992	48	48	200	46%	2015	14%
15	Uglašen ₁₃ ²	1987	50	50	200	47%	2015	7%
16	Uglašen ₁₄ ²	1966	49	49	200	45%	2015	8%
17	Uglašen ₁₅ ²	1963	48	48	200	45%	2015	11%
18	Uglašen ₁₆ ²	1949	49	49	200	43%	2015	17%
19	Uglašen ₁₇ ²	1945	49	49	200	42%	2015	11%
20	Rybka1900	1943	19	19	1050	44%	1992	10%
21	Uglašen ₁₈ ²	1942	50	50	200	42%	2015	8%
22	Uglašen ₁₉ ²	1942	50	50	200	42%	2015	10%
23	Uglašen ₂₀ ²	1936	49	49	200	42%	2015	14%
24	Začetni ²	1853	51	51	200	34%	2015	10%
25	Rybka1700	1705	23	23	1050	17%	1992	9%

(i) 9. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	26	26	1050	86%	1964	7%
2	Rybka ₂₁₀₀	2178	22	22	1050	75%	1964	8%
3	Uglašen ₁ ²	2071	48	48	200	51%	2059	9%
4	Uglašen ₂ ²	2054	48	48	200	49%	2059	8%
5	Uglašen ₃ ²	2047	48	48	200	48%	2059	9%
6	Rybka ₁₉₀₀	2018	19	19	1050	57%	1964	13%
7	Uglašen ₄ ²	2009	49	49	200	44%	2059	10%
8	Uglašen ₅ ²	2001	49	49	200	44%	2059	10%
9	Uglašen ₆ ²	1999	49	49	200	44%	2059	9%
10	Uglašen ₇ ²	1999	48	48	200	43%	2059	10%
11	Uglašen ₈ ²	1990	48	48	200	42%	2059	9%
12	Uglašen ₉ ²	1988	49	49	200	42%	2059	6%
13	Začetni ²	1980	49	49	200	41%	2059	10%
14	Uglašen ₁₀ ²	1976	48	48	200	42%	2059	13%
15	Uglašen ₁₁ ²	1974	49	49	200	41%	2059	9%
16	Uglašen ₁₂ ²	1966	49	49	200	39%	2059	10%
17	Uglašen ₁₃ ²	1959	49	49	200	39%	2059	12%
18	Uglašen ₁₄ ²	1956	49	49	200	39%	2059	9%
19	Uglašen ₁₅ ²	1914	51	51	200	34%	2059	12%
20	Uglašen ₁₆ ²	1905	52	52	200	34%	2059	7%
21	Uglašen ₁₇ ²	1896	51	51	200	32%	2059	11%
22	Uglašen ₁₈ ²	1890	51	51	200	33%	2059	12%
23	Uglašen ₁₉ ²	1834	51	51	200	28%	2059	19%
24	Uglašen ₂₀ ²	1830	54	54	200	27%	2059	10%
25	Rybka ₁₇₀₀	1740	21	21	1050	23%	1964	12%

(j) 10. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	25	25	1050	87%	1968	9%
2	Rybka ₂₁₀₀	2087	19	19	1050	65%	1968	15%
3	Uglašen ₁ ²	2040	48	48	200	56%	1997	17%
4	Uglašen ₂ ²	2039	50	50	200	54%	1997	6%
5	Uglašen ₃ ²	2033	48	48	200	53%	1997	15%
6	Uglašen ₄ ²	2031	47	47	200	53%	1997	18%
7	Uglašen ₅ ²	2007	48	48	200	51%	1997	12%
8	Uglašen ₆ ²	2006	47	47	200	51%	1997	17%
9	Uglašen ₇ ²	1998	49	49	200	50%	1997	13%
10	Uglašen ₈ ²	1990	49	49	200	49%	1995	14%
11	Uglašen ₉ ²	1984	48	48	200	48%	1997	14%
12	Uglašen ₁₀ ²	1983	48	48	200	48%	1997	14%
13	Uglašen ₁₁ ²	1978	49	49	200	49%	1997	11%
14	Uglašen ₁₂ ²	1969	49	49	200	47%	1997	8%
15	Uglašen ₁₃ ²	1967	49	49	200	47%	1997	10%
16	Uglašen ₁₄ ²	1963	49	49	200	47%	1997	13%
17	Uglašen ₁₅ ²	1942	49	49	200	44%	1997	10%
18	Uglašen ₁₆ ²	1935	48	48	200	43%	1997	15%
19	Uglašen ₁₇ ²	1934	48	48	200	44%	1997	12%
20	Uglašen ₁₈ ²	1932	48	48	200	42%	1997	17%
21	Rybka ₁₉₀₀	1918	19	19	1050	43%	1968	17%
22	Začetni ²	1891	50	50	200	39%	1997	11%
23	Uglašen ₁₉ ²	1869	51	51	200	37%	1997	10%
24	Uglašen ₂₀ ²	1846	52	52	200	35%	1997	11%
25	Rybka ₁₇₀₀	1681	24	24	1050	17%	1968	8%

Rezultati poskusov uglaševanja ocenitvene funkcije brez uporabe zgodovinskega mehanizma na ozkih intervalih. Podrobnejši opis poskusa je na str. 59.

Tabela 5.4: Rezultati uglaševanja brez zgodovinskega mehanizma na ozkih intervalih

(a) 1. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	30	30	1050	91%	1898	6%
2	Rybka ₂₁₀₀	2147	23	23	1050	79%	1898	8%
3	Rybka ₁₉₀₀	2033	21	21	1050	66%	1898	9%
4	Uglašen ₁ ^{10*}	1975	49	49	200	41%	2053	8%
5	Uglašen ₂ ^{10*}	1967	48	48	200	39%	2053	14%
6	Uglašen ₃ ^{10*}	1943	50	50	200	38%	2053	8%
7	Uglašen ₄ ^{10*}	1940	49	49	200	37%	2053	12%
8	Uglašen ₅ ^{10*}	1932	50	50	200	36%	2053	9%
9	Uglašen ₆ ^{10*}	1919	52	52	200	35%	2053	8%
10	Uglašen ₇ ^{10*}	1913	52	52	200	35%	2053	5%
11	Uglašen ₈ ^{10*}	1911	50	50	199	34%	2054	10%
12	Uglašen ₉ ^{10*}	1910	51	51	200	36%	2053	6%
13	Uglašen ₁₀ ^{10*}	1902	51	51	199	34%	2052	6%
14	Uglašen ₁₁ ^{10*}	1899	52	52	200	35%	2053	6%
15	Uglašen ₁₂ ^{10*}	1899	51	51	200	34%	2053	6%
16	Uglašen ₁₃ ^{10*}	1896	52	52	200	33%	2053	7%
17	Uglašen ₁₄ ^{10*}	1891	52	52	200	33%	2053	4%
18	Uglašen ₁₅ ^{10*}	1890	51	51	200	34%	2053	12%
19	Uglašen ₁₆ ^{10*}	1886	52	52	200	33%	2053	7%
20	Začetni ^{10*}	1875	54	54	200	32%	2053	5%
21	Uglašen ₁₇ ^{10*}	1869	52	52	200	31%	2053	6%
22	Uglašen ₁₈ ^{10*}	1859	53	53	200	30%	2053	7%
23	Uglašen ₁₉ ^{10*}	1828	55	55	200	28%	2053	4%
24	Uglašen ₂₀ ^{10*}	1760	57	57	200	20%	2053	11%
25	Rybka ₁₇₀₀	1732	21	21	1050	30%	1898	7%

(b) 2. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	34	34	1050	93%	1841	4%
2	Rybka ₂₁₀₀	2139	25	25	1050	83%	1841	6%
3	Rybka ₁₉₀₀	2035	21	21	1050	73%	1841	10%
4	Uglašen ₁ ^{10*}	1907	52	52	200	35%	2044	6%
5	Uglašen ₂ ^{10*}	1906	50	50	200	35%	2044	10%
6	Uglašen ₃ ^{10*}	1896	51	51	200	34%	2044	7%
7	Uglašen ₄ ^{10*}	1895	52	52	200	35%	2046	7%
8	Uglašen ₅ ^{10*}	1893	51	51	200	33%	2044	9%
9	Uglašen ₆ ^{10*}	1864	51	51	200	32%	2044	10%
10	Uglašen ₇ ^{10*}	1864	52	52	200	32%	2044	7%
11	Uglašen ₈ ^{10*}	1860	53	53	200	31%	2044	6%
12	Uglašen ₉ ^{10*}	1852	53	53	200	31%	2044	8%
13	Uglašen ₁₀ ^{10*}	1852	52	52	200	30%	2044	12%
14	Uglašen ₁₁ ^{10*}	1842	52	52	200	29%	2044	9%
15	Uglašen ₁₂ ^{10*}	1832	54	54	200	28%	2044	5%
16	Uglašen ₁₃ ^{10*}	1828	55	55	200	28%	2044	7%
17	Uglašen ₁₄ ^{10*}	1823	55	55	200	28%	2044	5%
18	Uglašen ₁₅ ^{10*}	1816	54	54	200	27%	2044	10%
19	Uglašen ₁₆ ^{10*}	1799	57	57	200	27%	2044	3%
20	Začetni ^{10*}	1788	52	52	200	26%	2034	5%
21	Uglašen ₁₇ ^{10*}	1787	55	55	200	26%	2044	9%
22	Uglašen ₁₈ ^{10*}	1786	56	56	200	24%	2044	7%
23	Uglašen ₁₉ ^{10*}	1786	55	55	200	25%	2044	10%
24	Uglašen ₂₀ ^{10*}	1783	55	55	200	25%	2044	10%
25	Rybka ₁₇₀₀	1701	21	21	1050	33%	1841	10%

(c) 3. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	26	26	1050	87%	1955	8%
2	Rybka ₂₁₀₀	2140	21	21	1050	72%	1955	9%
3	Uglašen ^{10*} ₁	2038	47	47	200	48%	2052	11%
4	Uglašen ^{10*} ₂	2014	49	49	200	46%	2052	7%
5	Uglašen ^{10*} ₃	2014	47	47	200	45%	2052	11%
6	Uglašen ^{10*} ₄	2012	47	47	200	46%	2052	12%
7	Uglašen ^{10*} ₅	1999	48	48	200	44%	2052	9%
8	Uglašen ^{10*} ₆	1989	48	48	200	44%	2052	7%
9	Rybka ₁₉₀₀	1988	20	20	1050	54%	1955	11%
10	Uglašen ^{10*} ₇	1978	49	49	200	42%	2052	9%
11	Uglašen ^{10*} ₈	1977	48	48	200	41%	2052	10%
12	Uglašen ^{10*} ₉	1974	47	47	200	41%	2052	12%
13	Uglašen ^{10*} ₁₀	1970	48	48	200	42%	2052	11%
14	Uglašen ^{10*} ₁₁	1947	49	49	200	39%	2052	7%
15	Uglašen ^{10*} ₁₂	1946	49	49	200	38%	2052	9%
16	Uglašen ^{10*} ₁₃	1940	48	48	200	38%	2052	12%
17	Uglašen ^{10*} ₁₄	1937	49	49	200	38%	2052	5%
18	Uglašen ^{10*} ₁₅	1933	49	49	200	37%	2052	7%
19	Uglašen ^{10*} ₁₆	1925	49	49	200	37%	2052	9%
20	Uglašen ^{10*} ₁₇	1903	49	49	200	34%	2052	12%
21	Uglašen ^{10*} ₁₈	1896	52	52	200	33%	2053	6%
22	Začetni ^{10*}	1896	51	51	200	33%	2053	7%
23	Uglašen ^{10*} ₁₉	1894	50	50	200	34%	2052	9%
24	Uglašen ^{10*} ₂₀	1883	51	51	200	32%	2052	10%
25	Rybka ₁₇₀₀	1780	21	21	1050	29%	1955	8%

(d) 4. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	32	32	1050	92%	1872	5%
2	Rybka ₂₁₀₀	2123	23	23	1050	79%	1872	9%
3	Uglašen ^{10*} ₁	2002	46	46	200	45%	2038	16%
4	Uglašen ^{10*} ₂	1978	48	48	200	44%	2038	8%
5	Uglašen ^{10*} ₃	1959	48	48	200	42%	2038	8%
6	Rybka ₁₉₀₀	1949	20	20	1050	60%	1872	13%
7	Uglašen ^{10*} ₄	1933	48	48	200	38%	2038	9%
8	Uglašen ^{10*} ₅	1914	49	49	200	36%	2039	8%
9	Uglašen ^{10*} ₆	1904	48	48	200	36%	2038	16%
10	Uglašen ^{10*} ₇	1892	49	49	200	35%	2038	10%
11	Uglašen ^{10*} ₈	1887	49	49	200	34%	2038	9%
12	Začetni ^{10*}	1885	50	50	200	33%	2038	10%
13	Uglašen ^{10*} ₉	1869	51	51	200	32%	2038	10%
14	Uglašen ^{10*} ₁₀	1862	50	50	200	32%	2038	11%
15	Uglašen ^{10*} ₁₁	1856	52	52	200	31%	2038	8%
16	Uglašen ^{10*} ₁₂	1850	52	52	200	31%	2038	6%
17	Uglašen ^{10*} ₁₃	1850	52	52	200	32%	2038	6%
18	Uglašen ^{10*} ₁₄	1847	49	49	200	31%	2038	13%
19	Uglašen ^{10*} ₁₅	1828	52	52	200	28%	2038	9%
20	Uglašen ^{10*} ₁₆	1824	52	52	200	27%	2038	13%
21	Uglašen ^{10*} ₁₇	1806	54	54	200	27%	2038	7%
22	Uglašen ^{10*} ₁₈	1802	54	54	200	26%	2038	4%
23	Uglašen ^{10*} ₁₉	1785	54	54	200	26%	2038	7%
24	Rybka ₁₇₀₀	1778	20	20	1050	38%	1872	10%
25	Uglašen ^{10*} ₂₀	1776	54	54	200	24%	2038	11%

(e) 5. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	26	26	1050	89%	1944	9%
2	Rybka ₂₁₀₀	2170	21	21	1050	77%	1944	12%
3	Uglašen ^{10*} ₁	2016	48	48	200	48%	2024	21%
4	Uglašen ^{10*} ₂	2010	49	49	200	48%	2024	14%
5	Uglašen ^{10*} ₃	2002	49	49	200	46%	2024	12%
6	Uglašen ^{10*} ₄	2000	49	49	200	46%	2024	15%
7	Uglašen ^{10*} ₅	1992	49	49	200	45%	2024	8%
8	Uglašen ^{10*} ₆	1988	50	50	200	46%	2024	7%
9	Uglašen ^{10*} ₇	1986	50	50	200	45%	2023	11%
10	Uglašen ^{10*} ₈	1976	50	50	200	43%	2024	10%
11	Uglašen ^{10*} ₉	1963	50	50	200	42%	2024	15%
12	Uglašen ^{10*} ₁₀	1958	50	50	200	42%	2024	9%
13	Uglašen ^{10*} ₁₁	1955	50	50	200	42%	2024	8%
14	Uglašen ^{10*} ₁₂	1955	50	50	200	42%	2024	11%
15	Rybka ₁₉₀₀	1941	19	19	1050	50%	1944	14%
16	Uglašen ^{10*} ₁₃	1939	49	49	200	41%	2024	10%
17	Uglašen ^{10*} ₁₄	1923	52	52	200	39%	2024	5%
18	Uglašen ^{10*} ₁₅	1919	50	50	200	40%	2024	12%
19	Uglašen ^{10*} ₁₆	1910	50	50	200	38%	2024	10%
20	Uglašen ^{10*} ₁₇	1908	49	49	200	38%	2024	17%
21	Uglašen ^{10*} ₁₈	1900	50	50	200	37%	2024	16%
22	Uglašen ^{10*} ₁₉	1874	53	53	200	34%	2026	11%
23	Uglašen ^{10*} ₂₀	1854	53	53	200	33%	2024	7%
24	Začetni ^{10*}	1800	54	54	200	28%	2025	5%
25	Rybka ₁₇₀₀	1686	23	23	1050	20%	1944	8%

(f) 6. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	26	26	1050	86%	1964	6%
2	Rybka ₂₁₀₀	2150	21	21	1050	73%	1964	11%
3	Uglašen ^{10*} ₁	2046	48	48	200	51%	2033	9%
4	Uglašen ^{10*} ₂	2027	49	49	200	49%	2033	11%
5	Uglašen ^{10*} ₃	2020	49	49	200	48%	2033	8%
6	Uglašen ^{10*} ₄	2018	49	49	200	48%	2033	11%
7	Uglašen ^{10*} ₅	2018	49	49	200	49%	2033	8%
8	Uglašen ^{10*} ₆	2013	48	48	200	48%	2033	9%
9	Uglašen ^{10*} ₇	2009	48	48	200	49%	2033	19%
10	Uglašen ^{10*} ₈	2001	49	49	200	46%	2033	8%
11	Uglašen ^{10*} ₉	1995	49	49	200	45%	2033	8%
12	Uglašen ^{10*} ₁₀	1984	48	48	200	45%	2033	13%
13	Uglašen ^{10*} ₁₁	1979	49	49	200	44%	2033	8%
14	Uglašen ^{10*} ₁₂	1973	50	50	200	43%	2033	7%
15	Rybka ₁₉₀₀	1959	19	19	1050	49%	1964	11%
16	Uglašen ^{10*} ₁₃	1948	49	49	200	40%	2034	11%
17	Uglašen ^{10*} ₁₄	1945	49	49	200	40%	2033	11%
18	Uglašen ^{10*} ₁₅	1941	49	49	200	41%	2033	10%
19	Uglašen ^{10*} ₁₆	1931	48	48	200	39%	2033	16%
20	Uglašen ^{10*} ₁₇	1900	50	50	200	36%	2033	9%
21	Uglašen ^{10*} ₁₈	1898	52	52	200	36%	2033	3%
22	Uglašen ^{10*} ₁₉	1898	50	50	200	36%	2033	11%
23	Začetni ^{10*}	1858	54	54	200	32%	2034	4%
24	Uglašen ^{10*} ₂₀	1843	53	53	200	31%	2033	6%
25	Rybka ₁₇₀₀	1721	22	22	1050	22%	1964	9%

(g) 7. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	34	34	1050	94%	1829	4%
2	Rybka ₂₁₀₀	2140	24	24	1050	84%	1829	8%
3	Uglašen ₁ ^{10*}	1988	48	48	200	45%	2028	12%
4	Rybka ₁₉₀₀	1921	20	20	1050	61%	1829	10%
5	Začetni ₁ ^{10*}	1920	49	49	200	40%	2028	7%
6	Uglašen ₂ ^{10*}	1883	50	50	200	35%	2028	5%
7	Uglašen ₃ ^{10*}	1878	51	51	200	35%	2028	8%
8	Uglašen ₄ ^{10*}	1871	51	51	200	34%	2028	6%
9	Uglašen ₅ ^{10*}	1861	50	50	200	32%	2028	10%
10	Uglašen ₆ ^{10*}	1856	52	52	200	32%	2028	8%
11	Uglašen ₇ ^{10*}	1850	51	51	200	32%	2028	8%
12	Uglašen ₈ ^{10*}	1844	51	51	200	31%	2028	13%
13	Uglašen ₉ ^{10*}	1831	51	51	200	30%	2028	8%
14	Uglašen ₁₀ ^{10*}	1829	52	52	200	30%	2028	5%
15	Uglašen ₁₁ ^{10*}	1824	51	51	200	29%	2028	16%
16	Uglašen ₁₂ ^{10*}	1814	51	51	200	30%	2027	13%
17	Uglašen ₁₃ ^{10*}	1806	53	53	200	28%	2028	8%
18	Uglašen ₁₄ ^{10*}	1796	54	54	200	27%	2028	4%
19	Uglašen ₁₅ ^{10*}	1793	51	51	200	26%	2028	14%
20	Uglašen ₁₆ ^{10*}	1792	54	54	200	26%	2028	6%
21	Uglašen ₁₇ ^{10*}	1775	52	52	200	25%	2028	15%
22	Rybka ₁₇₀₀	1753	19	19	1050	40%	1829	11%
23	Uglašen ₁₈ ^{10*}	1745	56	56	200	23%	2028	4%
24	Uglašen ₁₉ ^{10*}	1744	56	56	200	23%	2028	5%
25	Uglašen ₂₀ ^{10*}	1713	57	57	200	20%	2028	10%

(i) 9. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	37	37	1050	95%	1800	4%
2	Rybka ₂₁₀₀	2112	24	24	1050	84%	1800	8%
3	Rybka ₁₉₀₀	1930	20	20	1050	66%	1800	12%
4	Uglašen ₁ ^{10*}	1908	50	50	200	40%	2010	9%
5	Uglašen ₂ ^{10*}	1906	50	50	200	39%	2012	5%
6	Uglašen ₃ ^{10*}	1878	51	51	200	37%	2010	6%
7	Uglašen ₄ ^{10*}	1831	50	50	200	32%	2010	16%
8	Uglašen ₅ ^{10*}	1826	51	51	200	32%	2010	8%
9	Uglašen ₆ ^{10*}	1825	51	51	200	31%	2010	13%
10	Uglašen ₇ ^{10*}	1824	52	52	200	31%	2010	10%
11	Začetni ₁ ^{10*}	1823	51	51	200	30%	2012	14%
12	Uglašen ₈ ^{10*}	1816	51	51	200	31%	2010	11%
13	Uglašen ₉ ^{10*}	1814	52	52	200	30%	2010	8%
14	Uglašen ₁₀ ^{10*}	1808	52	52	200	30%	2010	11%
15	Uglašen ₁₁ ^{10*}	1807	53	53	200	30%	2010	7%
16	Uglašen ₁₂ ^{10*}	1803	54	54	200	30%	2010	6%
17	Uglašen ₁₃ ^{10*}	1802	53	53	200	29%	2010	7%
18	Uglašen ₁₄ ^{10*}	1800	53	53	200	29%	2010	5%
19	Uglašen ₁₅ ^{10*}	1797	52	52	200	28%	2010	12%
20	Uglašen ₁₆ ^{10*}	1754	55	55	200	25%	2010	9%
21	Uglašen ₁₇ ^{10*}	1708	57	57	200	22%	2010	6%
22	Uglašen ₁₈ ^{10*}	1704	57	57	200	21%	2010	8%
23	Uglašen ₁₉ ^{10*}	1703	58	58	200	22%	2010	5%
24	Rybka ₁₇₀₀	1699	20	20	1050	38%	1800	9%
25	Uglašen ₂₀ ^{10*}	1670	61	61	200	20%	2010	4%

(h) 8. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	33	33	1050	93%	1848	5%
2	Rybka ₂₁₀₀	2103	23	23	1050	79%	1848	8%
3	Rybka ₁₉₀₀	1980	20	20	1050	67%	1848	15%
4	Uglašen ₁ ^{10*}	1952	49	49	200	43%	2018	10%
5	Uglašen ₂ ^{10*}	1924	49	49	200	39%	2018	11%
6	Uglašen ₃ ^{10*}	1916	49	49	200	38%	2018	12%
7	Uglašen ₄ ^{10*}	1913	49	49	200	38%	2018	11%
8	Uglašen ₅ ^{10*}	1891	50	50	200	37%	2018	12%
9	Uglašen ₆ ^{10*}	1890	51	51	200	36%	2018	8%
10	Uglašen ₇ ^{10*}	1881	49	49	200	36%	2018	11%
11	Uglašen ₈ ^{10*}	1866	50	50	200	34%	2018	13%
12	Začetni ₁ ^{10*}	1862	51	51	200	33%	2018	10%
13	Uglašen ₉ ^{10*}	1849	52	52	200	32%	2018	7%
14	Uglašen ₁₀ ^{10*}	1846	51	51	200	32%	2018	10%
15	Uglašen ₁₁ ^{10*}	1845	51	51	200	32%	2018	13%
16	Uglašen ₁₂ ^{10*}	1833	51	51	200	31%	2018	10%
17	Uglašen ₁₃ ^{10*}	1830	53	53	200	31%	2018	5%
18	Uglašen ₁₄ ^{10*}	1830	53	53	200	31%	2018	7%
19	Uglašen ₁₅ ^{10*}	1820	52	52	200	31%	2018	11%
20	Uglašen ₁₆ ^{10*}	1819	51	51	200	31%	2018	14%
21	Uglašen ₁₇ ^{10*}	1807	54	54	200	29%	2018	7%
22	Uglašen ₁₈ ^{10*}	1798	52	52	200	27%	2018	10%
23	Uglašen ₁₉ ^{10*}	1739	55	55	200	23%	2018	14%
24	Uglašen ₂₀ ^{10*}	1701	58	58	200	20%	2018	8%
25	Rybka ₁₇₀₀	1691	21	21	1050	31%	1848	12%

(j) 10. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	25	25	1050	88%	1959	10%
2	Rybka ₂₁₀₀	2143	21	21	1050	73%	1959	11%
3	Uglašen ₁ ^{10*}	2041	47	47	200	49%	2041	17%
4	Uglašen ₂ ^{10*}	2029	48	48	200	48%	2041	17%
5	Uglašen ₃ ^{10*}	2007	48	48	200	45%	2043	9%
6	Uglašen ₄ ^{10*}	2004	46	46	200	46%	2041	19%
7	Uglašen ₅ ^{10*}	2001	48	48	200	46%	2041	9%
8	Rybka ₁₉₀₀	1980	19	19	1050	53%	1959	14%
9	Uglašen ₆ ^{10*}	1979	49	49	200	43%	2041	8%
10	Uglašen ₇ ^{10*}	1976	48	48	200	43%	2041	8%
11	Uglašen ₈ ^{10*}	1973	48	48	200	42%	2041	12%
12	Uglašen ₉ ^{10*}	1964	50	50	200	40%	2041	12%
13	Uglašen ₁₀ ^{10*}	1963	49	49	200	41%	2041	9%
14	Uglašen ₁₁ ^{10*}	1953	49	49	200	40%	2041	11%
15	Uglašen ₁₂ ^{10*}	1949	49	49	200	40%	2041	8%
16	Uglašen ₁₃ ^{10*}	1948	48	48	200	39%	2041	14%
17	Začetni ₁ ^{10*}	1946	49	49	200	39%	2043	11%
18	Uglašen ₁₄ ^{10*}	1943	48	48	200	39%	2041	13%
19	Uglašen ₁₅ ^{10*}	1932	49	49	200	37%	2041	12%
20	Uglašen ₁₆ ^{10*}	1919	51	51	200	37%	2041	7%
21	Uglašen ₁₇ ^{10*}	1917	50	50	200	37%	2041	3%
22	Uglašen ₁₈ ^{10*}	1908	50	50	200	36%	2041	7%
23	Uglašen ₁₉ ^{10*}	1903	50	50	200	35%	2041	12%
24	Uglašen ₂₀ ^{10*}	1882	49	49	200	35%	2041	17%
25	Rybka ₁₇₀₀	1742	22	22	1050	24%	1959	9%

Rezultati poskusov uglaševanja ocenitvene funkcije s pomočjo evolucijskega algoritma. Podrobnejši opis poskusa je na str. 62.

Tabela 5.5: Rezultati uglaševanja z evolucijskim algoritmom

(a) 1. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	24	24	1050	84%	1992	7%
2	Rybka ₂₁₀₀	2148	20	20	1050	70%	1992	14%
3	Uglašen ^{10#} ₁	2075	49	49	200	55%	2029	8%
4	Uglašen ^{10#} ₂	2064	48	48	200	54%	2029	14%
5	Uglašen ^{10#} ₃	2055	49	49	200	52%	2029	9%
6	Uglašen ^{10#} ₄	2051	48	48	200	52%	2029	14%
7	Uglašen ^{10#} ₅	2034	49	49	200	50%	2029	7%
8	Uglašen ^{10#} ₆	2026	48	48	200	50%	2029	14%
9	Uglašen ^{10#} ₇	2023	48	48	200	49%	2029	12%
10	Uglašen ^{10#} ₈	2005	48	48	200	46%	2029	9%
11	Uglašen ^{10#} ₉	1998	49	49	200	45%	2029	9%
12	Uglašen ^{10#} ₁₀	1998	49	49	200	45%	2029	16%
13	Uglašen ^{10#} ₁₁	1996	47	47	200	46%	2029	21%
14	Uglašen ^{10#} ₁₂	1995	48	48	200	46%	2029	17%
15	Uglašen ^{10#} ₁₃	1982	50	50	200	44%	2029	7%
16	Uglašen ^{10#} ₁₄	1967	50	50	200	44%	2029	8%
17	Uglašen ^{10#} ₁₅	1967	48	48	200	45%	2029	15%
18	Rybka ₁₉₀₀	1961	19	19	1050	46%	1992	16%
19	Uglašen ^{10#} ₁₆	1953	49	49	200	43%	2029	10%
20	Uglašen ^{10#} ₁₇	1943	50	50	200	40%	2029	11%
21	Uglašen ^{10#} ₁₈	1943	49	49	200	40%	2029	10%
22	Uglašen ^{10#} ₁₉	1942	49	49	200	40%	2029	12%
23	Uglašen ^{10#} ₂₀	1934	49	49	200	40%	2029	12%
24	Začetni ^{10#}	1882	51	51	200	34%	2029	11%
25	Rybka ₁₇₀₀	1710	23	23	1050	18%	1992	9%

(b) 2. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	25	25	1050	86%	1977	10%
2	Rybka ₂₁₀₀	2119	20	20	1050	67%	1977	11%
3	Uglašen ^{10#} ₁	2101	49	49	200	58%	2020	9%
4	Uglašen ^{10#} ₂	2028	49	49	200	50%	2020	7%
5	Uglašen ^{10#} ₃	2027	49	49	200	50%	2020	4%
6	Uglašen ^{10#} ₄	2026	49	49	200	49%	2020	9%
7	Uglašen ^{10#} ₅	2024	47	47	200	50%	2020	16%
8	Uglašen ^{10#} ₆	2024	47	47	200	50%	2020	17%
9	Uglašen ^{10#} ₇	2019	49	49	200	50%	2020	10%
10	Uglašen ^{10#} ₈	2017	49	49	200	50%	2020	11%
11	Uglašen ^{10#} ₉	2016	48	48	200	50%	2020	12%
12	Uglašen ^{10#} ₁₀	2012	48	48	200	50%	2020	12%
13	Uglašen ^{10#} ₁₁	2006	49	49	200	48%	2020	9%
14	Uglašen ^{10#} ₁₂	1989	47	47	200	47%	2020	10%
15	Uglašen ^{10#} ₁₃	1973	47	47	200	45%	2020	18%
16	Uglašen ^{10#} ₁₄	1966	48	48	200	45%	2020	11%
17	Rybka ₁₉₀₀	1936	19	19	1050	45%	1977	13%
18	Uglašen ^{10#} ₁₅	1930	49	49	200	40%	2020	16%
19	Uglašen ^{10#} ₁₆	1925	48	48	200	40%	2020	16%
20	Uglašen ^{10#} ₁₇	1917	49	49	200	39%	2020	8%
21	Uglašen ^{10#} ₁₈	1911	49	49	200	39%	2020	16%
22	Uglašen ^{10#} ₁₉	1894	51	51	200	36%	2020	11%
23	Uglašen ^{10#} ₂₀	1877	51	51	200	35%	2020	9%
24	Začetni ^{10#}	1837	51	51	200	32%	2020	7%
25	Rybka ₁₇₀₀	1727	22	22	1050	21%	1977	11%

(c) 3. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	24	24	1050	85%	1986	9%
2	Rybka2100	2145	20	20	1050	70%	1986	13%
3	Uglašen ^{10#} ₁	2061	47	47	200	53%	2031	17%
4	Uglašen ^{10#} ₂	2043	47	47	200	52%	2031	16%
5	Uglašen ^{10#} ₃	2030	49	49	200	50%	2031	11%
6	Uglašen ^{10#} ₄	2027	47	47	200	50%	2031	16%
7	Uglašen ^{10#} ₅	2026	48	48	200	50%	2031	13%
8	Uglašen ^{10#} ₆	2021	48	48	200	49%	2031	10%
9	Uglašen ^{10#} ₇	2017	48	48	200	48%	2031	9%
10	Uglašen ^{10#} ₈	2016	48	48	200	47%	2031	10%
11	Uglašen ^{10#} ₉	2010	48	48	200	47%	2031	12%
12	Uglašen ^{10#} ₁₀	1992	47	47	200	45%	2031	15%
13	Uglašen ^{10#} ₁₁	1989	47	47	200	47%	2031	15%
14	Uglašen ^{10#} ₁₂	1983	46	46	200	46%	2031	18%
15	Uglašen ^{10#} ₁₃	1977	48	48	200	45%	2031	11%
16	Uglašen ^{10#} ₁₄	1975	48	48	200	44%	2031	13%
17	Uglašen ^{10#} ₁₅	1970	48	48	200	44%	2031	11%
18	Uglašen ^{10#} ₁₆	1958	48	48	200	43%	2031	9%
19	Uglašen ^{10#} ₁₇	1955	50	50	200	41%	2031	10%
20	Uglašen ^{10#} ₁₈	1949	50	50	200	41%	2031	7%
21	Uglašen ^{10#} ₁₉	1940	50	50	200	41%	2031	7%
22	Uglašen ^{10#} ₂₀	1933	49	49	200	40%	2031	13%
23	Rybka1900	1909	19	19	1050	40%	1986	14%
24	Začetni ^{10#}	1829	53	53	200	30%	2031	4%
25	Rybka1700	1769	21	21	1050	24%	1986	10%

(d) 4. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	23	23	1050	83%	2013	11%
2	Uglašen ^{10#} ₁	2150	49	49	200	64%	2018	12%
3	Rybka2100	2127	20	20	1050	65%	2013	14%
4	Uglašen ^{10#} ₂	2125	49	49	200	62%	2018	12%
5	Uglašen ^{10#} ₃	2102	50	50	200	58%	2018	8%
6	Uglašen ^{10#} ₄	2073	49	49	200	56%	2018	13%
7	Uglašen ^{10#} ₅	2059	48	48	200	54%	2018	12%
8	Uglašen ^{10#} ₆	2056	48	48	200	54%	2018	10%
9	Uglašen ^{10#} ₇	2042	48	48	200	51%	2018	12%
10	Uglašen ^{10#} ₈	2036	49	49	200	51%	2018	12%
11	Uglašen ^{10#} ₉	2034	49	49	200	51%	2018	9%
12	Uglašen ^{10#} ₁₀	2025	49	49	200	50%	2018	10%
13	Uglašen ^{10#} ₁₁	2024	48	48	200	50%	2018	12%
14	Uglašen ^{10#} ₁₂	2011	48	48	200	48%	2018	15%
15	Uglašen ^{10#} ₁₃	1971	49	49	200	44%	2018	10%
16	Uglašen ^{10#} ₁₄	1969	49	49	200	45%	2018	8%
17	Uglašen ^{10#} ₁₅	1964	47	47	200	45%	2018	19%
18	Uglašen ^{10#} ₁₆	1959	48	48	200	43%	2018	16%
19	Uglašen ^{10#} ₁₇	1958	50	50	200	44%	2018	11%
20	Uglašen ^{10#} ₁₈	1934	49	49	200	41%	2018	14%
21	Uglašen ^{10#} ₁₉	1933	48	48	200	41%	2018	12%
22	Uglašen ^{10#} ₂₀	1928	51	51	200	40%	2018	10%
23	Rybka1900	1928	19	19	1050	39%	2013	14%
24	Začetni ^{10#}	1924	49	49	200	40%	2018	10%
25	Rybka1700	1717	24	24	1050	17%	2013	7%

(e) 5. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	22	22	1050	79%	2058	12%
2	Uglašen ^{10#} ₁	2140	48	48	200	61%	2038	10%
3	Uglašen ^{10#} ₂	2129	47	47	200	61%	2038	15%
4	Rybka2100	2127	19	19	1050	59%	2058	18%
5	Uglašen ^{10#} ₃	2108	47	47	200	57%	2038	14%
6	Uglašen ^{10#} ₄	2104	48	48	200	58%	2038	11%
7	Uglašen ^{10#} ₅	2096	48	48	200	56%	2038	12%
8	Uglašen ^{10#} ₆	2094	48	48	200	56%	2038	14%
9	Uglašen ^{10#} ₇	2083	48	48	200	54%	2038	12%
10	Uglašen ^{10#} ₈	2082	47	47	200	55%	2038	16%
11	Uglašen ^{10#} ₉	2080	48	48	200	55%	2038	12%
12	Uglašen ^{10#} ₁₀	2066	47	47	200	53%	2038	17%
13	Uglašen ^{10#} ₁₁	2060	47	47	200	52%	2038	12%
14	Uglašen ^{10#} ₁₂	2052	48	48	200	51%	2038	10%
15	Uglašen ^{10#} ₁₃	2040	47	47	200	50%	2038	17%
16	Uglašen ^{10#} ₁₄	2033	47	47	200	49%	2038	17%
17	Uglašen ^{10#} ₁₅	2029	47	47	200	48%	2038	14%
18	Uglašen ^{10#} ₁₆	2029	46	46	200	49%	2038	15%
19	Uglašen ^{10#} ₁₇	2028	47	47	200	48%	2038	13%
20	Uglašen ^{10#} ₁₈	2025	47	47	200	48%	2038	13%
21	Uglašen ^{10#} ₁₉	2020	47	47	200	46%	2038	18%
22	Rybka1900	1970	19	19	1050	39%	2058	14%
23	Začetni ^{10#}	1969	49	49	200	42%	2038	8%
24	Uglašen ^{10#} ₂₀	1957	48	48	200	41%	2038	14%
25	Rybka1700	1755	24	24	1050	16%	2058	9%

(f) 6. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	23	23	1050	83%	2013	11%
2	Rybka2100	2140	20	20	1050	66%	2013	14%
3	Uglašen ^{10#} ₁	2131	47	47	200	59%	2043	15%
4	Uglašen ^{10#} ₂	2114	47	47	200	59%	2043	18%
5	Uglašen ^{10#} ₃	2092	47	47	200	55%	2043	11%
6	Uglašen ^{10#} ₄	2077	47	47	200	53%	2043	12%
7	Uglašen ^{10#} ₅	2076	46	46	200	53%	2041	15%
8	Uglašen ^{10#} ₆	2066	47	47	200	53%	2043	15%
9	Uglašen ^{10#} ₇	2064	47	47	200	53%	2043	14%
10	Uglašen ^{10#} ₈	2046	48	48	200	50%	2043	7%
11	Uglašen ^{10#} ₉	2032	46	46	200	49%	2043	18%
12	Uglašen ^{10#} ₁₀	2016	47	47	200	46%	2043	14%
13	Uglašen ^{10#} ₁₁	2006	47	47	200	46%	2043	12%
14	Uglašen ^{10#} ₁₂	2003	48	48	200	45%	2043	9%
15	Uglašen ^{10#} ₁₃	1997	47	47	200	44%	2043	17%
16	Uglašen ^{10#} ₁₄	1988	48	48	200	43%	2043	13%
17	Uglašen ^{10#} ₁₅	1986	48	48	200	44%	2043	11%
18	Uglašen ^{10#} ₁₆	1985	48	48	200	44%	2043	12%
19	Uglašen ^{10#} ₁₇	1984	47	47	200	44%	2043	14%
20	Uglašen ^{10#} ₁₈	1978	49	49	200	42%	2043	12%
21	Rybka1900	1973	19	19	1050	44%	2013	17%
22	Uglašen ^{10#} ₁₉	1963	49	49	200	41%	2043	11%
23	Uglašen ^{10#} ₂₀	1957	47	47	200	41%	2043	18%
24	Rybka1700	1759	23	23	1050	21%	2013	9%
25	Začetni ^{10#}	1717	58	58	200	20%	2043	7%

(g) 7. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	28	28	1050	90%	1914	7%
2	Rybka ₂₁₀₀	2113	21	21	1050	74%	1914	13%
3	Uglašen ^{10#} ₁	2048	48	48	200	53%	2019	13%
4	Uglašen ^{10#} ₂	2025	47	47	200	51%	2019	16%
5	Uglašen ^{10#} ₃	2011	49	49	200	49%	2019	7%
6	Uglašen ^{10#} ₄	1992	48	48	200	47%	2019	11%
7	Uglašen ^{10#} ₅	1979	49	49	200	47%	2019	12%
8	Uglašen ^{10#} ₆	1968	46	46	200	44%	2019	21%
9	Uglašen ^{10#} ₇	1949	48	48	200	42%	2019	15%
10	Rybka ₁₉₀₀	1943	19	19	1050	53%	1914	14%
11	Uglašen ^{10#} ₈	1939	49	49	200	42%	2019	10%
12	Uglašen ^{10#} ₉	1925	49	49	200	39%	2019	11%
13	Uglašen ^{10#} ₁₀	1923	48	48	200	39%	2019	11%
14	Uglašen ^{10#} ₁₁	1915	50	50	200	39%	2019	9%
15	Uglašen ^{10#} ₁₂	1897	50	50	200	38%	2019	6%
16	Uglašen ^{10#} ₁₃	1895	50	50	200	36%	2019	13%
17	Uglašen ^{10#} ₁₄	1893	49	49	200	36%	2019	15%
18	Uglašen ^{10#} ₁₅	1893	49	49	200	37%	2019	9%
19	Uglašen ^{10#} ₁₆	1878	49	49	200	36%	2019	17%
20	Uglašen ^{10#} ₁₇	1878	50	50	200	35%	2019	8%
21	Uglašen ^{10#} ₁₈	1874	50	50	200	35%	2019	10%
22	Uglašen ^{10#} ₁₉	1831	51	51	200	31%	2019	9%
23	Uglašen ^{10#} ₂₀	1807	54	54	200	29%	2019	7%
24	Rybka ₁₇₀₀	1721	21	21	1050	27%	1914	9%
25	Začetni ^{10#}	1681	61	61	200	19%	2019	4%

(h) 8. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	25	25	1050	86%	1974	9%
2	Rybka ₂₁₀₀	2091	19	19	1050	65%	1974	15%
3	Uglašen ^{10#} ₁	2045	48	48	200	54%	2007	15%
4	Uglašen ^{10#} ₂	2032	47	47	200	53%	2007	16%
5	Uglašen ^{10#} ₃	2027	48	48	200	52%	2007	11%
6	Uglašen ^{10#} ₄	2013	49	49	200	50%	2007	8%
7	Uglašen ^{10#} ₅	2012	48	48	200	50%	2007	13%
8	Uglašen ^{10#} ₆	2004	47	47	200	49%	2007	14%
9	Uglašen ^{10#} ₇	1995	48	48	200	49%	2007	11%
10	Začetni ^{10#}	1986	49	49	200	49%	2007	7%
11	Uglašen ^{10#} ₈	1981	49	49	200	48%	2007	9%
12	Uglašen ^{10#} ₉	1975	49	49	200	46%	2006	9%
13	Uglašen ^{10#} ₁₀	1973	48	48	200	48%	2007	15%
14	Uglašen ^{10#} ₁₁	1967	47	47	200	45%	2007	14%
15	Uglašen ^{10#} ₁₂	1955	48	48	200	44%	2007	13%
16	Uglašen ^{10#} ₁₃	1954	49	49	200	45%	2007	8%
17	Uglašen ^{10#} ₁₄	1953	49	49	200	44%	2007	9%
18	Uglašen ^{10#} ₁₅	1952	47	47	200	46%	2007	22%
19	Uglašen ^{10#} ₁₆	1947	50	50	200	44%	2007	7%
20	Uglašen ^{10#} ₁₇	1946	49	49	200	44%	2007	9%
21	Rybka ₁₉₀₀	1925	19	19	1050	44%	1974	12%
22	Uglašen ^{10#} ₁₈	1916	48	48	200	41%	2007	21%
23	Uglašen ^{10#} ₁₉	1911	47	47	200	39%	2007	22%
24	Uglašen ^{10#} ₂₀	1902	50	50	200	39%	2007	6%
25	Rybka ₁₇₀₀	1712	22	22	1050	19%	1974	12%

(i) 9. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	23	23	1050	82%	2021	10%
2	Rybka ₂₁₀₀	2151	20	20	1050	66%	2021	13%
3	Uglašen ^{10#} ₁	2140	49	49	200	61%	2034	9%
4	Uglašen ^{10#} ₂	2080	48	48	200	55%	2034	10%
5	Uglašen ^{10#} ₃	2077	48	48	200	55%	2034	16%
6	Uglašen ^{10#} ₄	2067	48	48	200	54%	2034	8%
7	Uglašen ^{10#} ₅	2066	49	49	200	54%	2034	6%
8	Uglašen ^{10#} ₆	2061	48	48	200	53%	2034	14%
9	Uglašen ^{10#} ₇	2045	49	49	200	51%	2034	10%
10	Uglašen ^{10#} ₈	2034	49	49	200	50%	2034	9%
11	Uglašen ^{10#} ₉	2021	49	49	200	49%	2034	10%
12	Uglašen ^{10#} ₁₀	2014	46	46	200	48%	2034	17%
13	Uglašen ^{10#} ₁₁	2006	47	47	200	47%	2034	16%
14	Uglašen ^{10#} ₁₂	2005	49	49	200	47%	2034	7%
15	Uglašen ^{10#} ₁₃	2005	49	49	200	46%	2034	9%
16	Uglašen ^{10#} ₁₄	1999	47	47	200	45%	2034	12%
17	Uglašen ^{10#} ₁₅	1992	48	48	200	46%	2034	11%
18	Uglašen ^{10#} ₁₆	1990	47	47	200	46%	2034	15%
19	Uglašen ^{10#} ₁₇	1989	47	47	200	45%	2034	20%
20	Uglašen ^{10#} ₁₈	1981	49	49	200	44%	2034	12%
21	Uglašen ^{10#} ₁₉	1975	48	48	200	44%	2034	14%
22	Uglašen ^{10#} ₂₀	1961	49	49	200	43%	2034	10%
23	Začetni ^{10#}	1928	49	49	200	40%	2034	11%
24	Rybka ₁₉₀₀	1916	20	20	1050	37%	2021	12%
25	Rybka ₁₇₀₀	1770	22	22	1050	20%	2021	12%

(j) 10. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	23	23	1050	83%	2016	10%
2	Rybka ₂₁₀₀	2116	19	19	1050	63%	2016	14%
3	Uglašen ^{10#} ₁	2113	49	49	200	59%	2030	10%
4	Uglašen ^{10#} ₂	2103	48	48	200	57%	2030	13%
5	Uglašen ^{10#} ₃	2100	47	47	200	58%	2030	12%
6	Uglašen ^{10#} ₄	2074	48	48	200	56%	2030	16%
7	Uglašen ^{10#} ₅	2045	47	47	200	52%	2030	15%
8	Uglašen ^{10#} ₆	2041	48	48	200	51%	2030	11%
9	Uglašen ^{10#} ₇	2038	47	47	200	51%	2030	15%
10	Uglašen ^{10#} ₈	2037	47	47	200	51%	2030	13%
11	Uglašen ^{10#} ₉	2035	48	48	200	50%	2030	8%
12	Uglašen ^{10#} ₁₀	2021	47	47	200	50%	2030	18%
13	Uglašen ^{10#} ₁₁	2017	47	47	200	50%	2030	16%
14	Uglašen ^{10#} ₁₂	2007	47	47	200	48%	2030	12%
15	Uglašen ^{10#} ₁₃	2005	45	45	200	47%	2030	24%
16	Uglašen ^{10#} ₁₄	2002	46	46	200	47%	2030	17%
17	Uglašen ^{10#} ₁₅	2000	46	46	200	49%	2030	19%
18	Uglašen ^{10#} ₁₆	1966	48	48	200	43%	2030	10%
19	Uglašen ^{10#} ₁₇	1963	48	48	200	43%	2030	15%
20	Uglašen ^{10#} ₁₈	1960	48	48	200	42%	2030	7%
21	Uglašen ^{10#} ₁₉	1949	48	48	200	41%	2030	13%
22	Rybka ₁₉₀₀	1948	19	19	1050	41%	2016	15%
23	Uglašen ^{10#} ₂₀	1930	49	49	200	41%	2030	8%
24	Začetni ^{10#}	1923	49	49	200	38%	2030	8%
25	Rybka ₁₇₀₀	1756	22	22	1050	19%	2016	14%

Rezultati poskusov uglaševanja ocenitvene funkcije na širših intervalih. Podrobnejši opis poskusa je na str. 68.

Tabela 5.6: Rezultati uglaševanja na širših intervalih

(a) 1. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	35	35	1050	94%	1801	3%
2	Rybka ₂₁₀₀	2132	25	25	1050	84%	1801	7%
3	Rybka ₁₉₀₀	1964	20	20	1050	68%	1801	11%
4	Uglašen ₁ ¹⁰⁺	1901	49	49	200	34%	2045	11%
5	Uglašen ₂ ¹⁰⁺	1894	49	49	200	34%	2045	10%
6	Uglašen ₃ ¹⁰⁺	1887	50	50	200	34%	2045	8%
7	Uglašen ₄ ¹⁰⁺	1875	49	49	200	33%	2045	15%
8	Uglašen ₅ ¹⁰⁺	1869	49	49	200	32%	2045	13%
9	Uglašen ₆ ¹⁰⁺	1860	50	50	200	31%	2045	11%
10	Uglašen ₇ ¹⁰⁺	1856	50	50	200	30%	2045	10%
11	Uglašen ₈ ¹⁰⁺	1851	52	52	200	31%	2045	7%
12	Uglašen ₉ ¹⁰⁺	1838	53	53	200	29%	2045	6%
13	Uglašen ₁₀ ¹⁰⁺	1836	53	53	200	29%	2045	4%
14	Uglašen ₁₁ ¹⁰⁺	1832	52	52	200	28%	2045	9%
15	Uglašen ₁₂ ¹⁰⁺	1825	53	53	200	28%	2045	5%
16	Uglašen ₁₃ ¹⁰⁺	1817	54	54	200	27%	2045	4%
17	Uglašen ₁₄ ¹⁰⁺	1807	53	53	200	26%	2045	7%
18	Uglašen ₁₅ ¹⁰⁺	1807	54	54	200	26%	2045	6%
19	Uglašen ₁₆ ¹⁰⁺	1805	53	53	200	26%	2045	6%
20	Uglašen ₁₇ ¹⁰⁺	1787	54	54	200	24%	2045	7%
21	Rybka ₁₇₀₀	1783	20	20	1050	46%	1801	9%
22	Uglašen ₁₈ ¹⁰⁺	1782	53	53	200	24%	2045	9%
23	Uglašen ₁₉ ¹⁰⁺	1765	56	56	200	22%	2045	6%
24	Uglašen ₂₀ ¹⁰⁺	1740	57	57	200	21%	2045	5%
25	Začetni ¹⁰⁺	1189	179	179	200	1%	2045	1%

(b) 2. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	28	28	1050	89%	1919	7%
2	Rybka ₂₁₀₀	2159	22	22	1050	78%	1919	9%
3	Uglašen ₁ ¹⁰⁺	2023	47	47	200	47%	2049	9%
4	Uglašen ₂ ¹⁰⁺	1989	47	47	200	44%	2049	10%
5	Uglašen ₃ ¹⁰⁺	1985	48	48	200	43%	2049	7%
6	Uglašen ₄ ¹⁰⁺	1968	49	49	200	42%	2049	4%
7	Uglašen ₅ ¹⁰⁺	1964	48	48	200	40%	2049	13%
8	Uglašen ₆ ¹⁰⁺	1949	48	48	200	40%	2049	9%
9	Uglašen ₇ ¹⁰⁺	1948	49	49	200	39%	2049	7%
10	Uglašen ₈ ¹⁰⁺	1946	49	49	200	39%	2049	7%
11	Uglašen ₉ ¹⁰⁺	1942	49	49	200	38%	2049	13%
12	Rybka ₁₉₀₀	1942	19	19	1050	53%	1919	13%
13	Uglašen ₁₀ ¹⁰⁺	1936	49	49	200	38%	2049	9%
14	Uglašen ₁₁ ¹⁰⁺	1930	47	47	200	38%	2049	15%
15	Uglašen ₁₂ ¹⁰⁺	1928	49	49	200	36%	2049	11%
16	Uglašen ₁₃ ¹⁰⁺	1923	50	50	200	37%	2049	8%
17	Uglašen ₁₄ ¹⁰⁺	1917	48	48	200	36%	2049	15%
18	Uglašen ₁₅ ¹⁰⁺	1917	49	49	200	36%	2049	12%
19	Uglašen ₁₆ ¹⁰⁺	1914	49	49	200	36%	2049	12%
20	Uglašen ₁₇ ¹⁰⁺	1882	52	52	200	33%	2049	9%
21	Uglašen ₁₈ ¹⁰⁺	1881	50	50	200	32%	2049	15%
22	Uglašen ₁₉ ¹⁰⁺	1861	49	49	200	31%	2049	14%
23	Uglašen ₂₀ ¹⁰⁺	1861	51	51	200	31%	2049	10%
24	Rybka ₁₇₀₀	1795	20	20	1050	35%	1919	11%
25	Začetni ¹⁰⁺	1639	63	63	200	13%	2049	8%

(c) 3. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	29	29	1050	90%	1887	5%
2	Rybka2100	2120	22	22	1050	77%	1887	10%
3	Uglašen ¹⁰⁺ ₁	1993	48	48	200	47%	2020	12%
4	Uglašen ¹⁰⁺ ₂	1967	50	50	200	45%	2020	3%
5	Uglašen ¹⁰⁺ ₃	1943	48	48	200	42%	2020	12%
6	Uglašen ¹⁰⁺ ₄	1942	48	48	200	41%	2020	11%
7	Uglašen ¹⁰⁺ ₅	1934	48	48	200	42%	2020	17%
8	Uglašen ¹⁰⁺ ₆	1926	49	49	200	40%	2020	9%
9	Rybka1900	1918	19	19	1050	53%	1887	12%
10	Uglašen ¹⁰⁺ ₇	1915	50	50	200	39%	2020	8%
11	Uglašen ¹⁰⁺ ₈	1908	50	50	200	39%	2020	7%
12	Uglašen ¹⁰⁺ ₉	1907	49	49	200	39%	2020	11%
13	Uglašen ¹⁰⁺ ₁₀	1903	50	50	200	38%	2020	6%
14	Uglašen ¹⁰⁺ ₁₁	1900	50	50	200	38%	2020	11%
15	Uglašen ¹⁰⁺ ₁₂	1899	49	49	200	38%	2020	12%
16	Uglašen ¹⁰⁺ ₁₃	1891	50	50	200	37%	2020	10%
17	Uglašen ¹⁰⁺ ₁₄	1891	49	49	200	38%	2020	11%
18	Uglašen ¹⁰⁺ ₁₅	1891	50	50	200	37%	2020	10%
19	Uglašen ¹⁰⁺ ₁₆	1891	50	50	200	36%	2020	7%
20	Uglašen ¹⁰⁺ ₁₇	1879	50	50	200	36%	2020	11%
21	Uglašen ¹⁰⁺ ₁₈	1862	51	51	200	34%	2020	9%
22	Uglašen ¹⁰⁺ ₁₉	1850	50	50	200	33%	2020	11%
23	Uglašen ¹⁰⁺ ₂₀	1828	52	52	200	31%	2020	8%
24	Rybka1700	1740	20	20	1050	32%	1887	10%
25	Začetni ¹⁰⁺	1500	77	77	200	9%	2020	2%

(d) 4. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	27	27	1049	88%	1934	8%
2	Rybka2100	2088	20	20	1050	69%	1934	14%
3	Uglašen ¹⁰⁺ ₁	2031	47	47	200	49%	2034	13%
4	Uglašen ¹⁰⁺ ₂	2012	47	47	200	48%	2034	10%
5	Uglašen ¹⁰⁺ ₃	2006	46	46	200	47%	2034	14%
6	Uglašen ¹⁰⁺ ₄	1993	47	47	200	46%	2034	11%
7	Uglašen ¹⁰⁺ ₅	1987	46	46	200	44%	2034	14%
8	Uglašen ¹⁰⁺ ₆	1976	47	47	200	43%	2034	12%
9	Uglašen ¹⁰⁺ ₇	1974	48	48	200	43%	2034	8%
10	Uglašen ¹⁰⁺ ₈	1967	46	46	200	42%	2034	18%
11	Rybka1900	1966	19	19	1050	54%	1934	15%
12	Uglašen ¹⁰⁺ ₉	1964	46	46	200	43%	2034	20%
13	Uglašen ¹⁰⁺ ₁₀	1964	46	46	200	43%	2034	18%
14	Uglašen ¹⁰⁺ ₁₁	1958	48	48	199	41%	2033	13%
15	Uglašen ¹⁰⁺ ₁₂	1933	47	47	200	39%	2034	15%
16	Uglašen ¹⁰⁺ ₁₃	1933	48	48	200	40%	2034	15%
17	Uglašen ¹⁰⁺ ₁₄	1932	47	47	200	40%	2034	14%
18	Uglašen ¹⁰⁺ ₁₅	1930	48	48	200	39%	2034	13%
19	Uglašen ¹⁰⁺ ₁₆	1925	48	48	200	39%	2034	12%
20	Uglašen ¹⁰⁺ ₁₇	1922	47	47	200	38%	2034	15%
21	Uglašen ¹⁰⁺ ₁₈	1872	51	51	200	32%	2034	8%
22	Uglašen ¹⁰⁺ ₁₉	1858	50	50	200	31%	2034	11%
23	Uglašen ¹⁰⁺ ₂₀	1851	49	49	200	31%	2034	12%
24	Rybka1700	1782	20	20	1050	31%	1934	14%
25	Začetni ¹⁰⁺	1628	64	64	200	14%	2034	5%

(e) 5. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	29	29	1050	91%	1876	7%
2	Rybka2100	2166	23	23	1050	81%	1876	8%
3	Rybka1900	2014	20	20	1050	65%	1876	9%
4	Uglašen ¹⁰⁺ ₁	1976	48	48	200	40%	2063	11%
5	Uglašen ¹⁰⁺ ₂	1968	49	49	200	39%	2063	7%
6	Uglašen ¹⁰⁺ ₃	1954	50	50	200	38%	2063	5%
7	Uglašen ¹⁰⁺ ₄	1946	50	50	200	37%	2063	8%
8	Uglašen ¹⁰⁺ ₅	1939	48	48	200	35%	2063	18%
9	Uglašen ¹⁰⁺ ₆	1934	50	50	200	36%	2063	7%
10	Uglašen ¹⁰⁺ ₇	1930	50	50	200	36%	2063	5%
11	Uglašen ¹⁰⁺ ₈	1927	50	50	200	36%	2063	7%
12	Uglašen ¹⁰⁺ ₉	1919	50	50	200	35%	2063	8%
13	Uglašen ¹⁰⁺ ₁₀	1911	50	50	200	33%	2063	13%
14	Uglašen ¹⁰⁺ ₁₁	1907	50	50	200	33%	2063	9%
15	Uglašen ¹⁰⁺ ₁₂	1902	49	49	200	33%	2063	14%
16	Uglašen ¹⁰⁺ ₁₃	1898	51	51	200	33%	2063	7%
17	Uglašen ¹⁰⁺ ₁₄	1885	53	53	200	32%	2063	5%
18	Uglašen ¹⁰⁺ ₁₅	1884	51	51	200	31%	2063	10%
19	Uglašen ¹⁰⁺ ₁₆	1882	52	52	200	31%	2063	8%
20	Uglašen ¹⁰⁺ ₁₇	1869	52	52	200	30%	2063	5%
21	Uglašen ¹⁰⁺ ₁₈	1847	53	53	200	28%	2063	9%
22	Uglašen ¹⁰⁺ ₁₉	1845	53	53	200	28%	2063	6%
23	Uglašen ¹⁰⁺ ₂₀	1836	54	54	200	27%	2063	4%
24	Rybka1700	1772	21	21	1050	37%	1876	7%
25	Začetni ¹⁰⁺	1239	158	158	200	1%	2063	1%

(f) 6. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka2300	2300	28	28	1050	90%	1911	7%
2	Rybka2100	2162	22	22	1050	79%	1911	10%
3	Uglašen ¹⁰⁺ ₁	1999	49	49	200	45%	2037	10%
4	Uglašen ¹⁰⁺ ₂	1990	50	50	200	45%	2037	10%
5	Uglašen ¹⁰⁺ ₃	1970	50	50	200	42%	2037	6%
6	Rybka1900	1957	19	19	1050	56%	1911	15%
7	Uglašen ¹⁰⁺ ₄	1956	50	50	200	41%	2037	9%
8	Uglašen ¹⁰⁺ ₅	1953	48	48	200	40%	2037	17%
9	Uglašen ¹⁰⁺ ₆	1952	49	49	200	41%	2037	9%
10	Uglašen ¹⁰⁺ ₇	1947	49	49	200	41%	2037	10%
11	Uglašen ¹⁰⁺ ₈	1929	49	49	200	39%	2037	12%
12	Uglašen ¹⁰⁺ ₉	1929	49	49	200	39%	2037	10%
13	Uglašen ¹⁰⁺ ₁₀	1923	49	49	200	37%	2037	17%
14	Uglašen ¹⁰⁺ ₁₁	1916	50	50	200	37%	2037	11%
15	Uglašen ¹⁰⁺ ₁₂	1916	51	51	200	38%	2037	6%
16	Uglašen ¹⁰⁺ ₁₃	1912	50	50	200	37%	2037	8%
17	Uglašen ¹⁰⁺ ₁₄	1912	49	49	200	36%	2037	13%
18	Uglašen ¹⁰⁺ ₁₅	1908	51	51	200	36%	2037	11%
19	Uglašen ¹⁰⁺ ₁₆	1904	51	51	200	36%	2037	7%
20	Uglašen ¹⁰⁺ ₁₇	1904	49	49	200	36%	2037	17%
21	Uglašen ¹⁰⁺ ₁₈	1880	50	50	200	33%	2037	12%
22	Uglašen ¹⁰⁺ ₁₉	1879	50	50	200	34%	2037	15%
23	Uglašen ¹⁰⁺ ₂₀	1875	50	50	200	34%	2037	12%
24	Rybka1700	1729	21	21	1050	28%	1911	10%
25	Začetni ¹⁰⁺	1572	70	70	200	12%	2037	3%

(g) 7. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	28	28	1049	89%	1917	6%
2	Rybka ₂₁₀₀	2175	22	22	1050	79%	1917	10%
3	Uglašen ₁ ¹⁰⁺	2030	47	47	200	47%	2062	13%
4	Uglašen ₂ ¹⁰⁺	2019	47	47	200	44%	2062	15%
5	Rybka ₁₉₀₀	2016	20	20	1050	62%	1917	13%
6	Uglašen ₃ ¹⁰⁺	1992	49	49	200	43%	2062	8%
7	Uglašen ₄ ¹⁰⁺	1981	49	49	200	41%	2062	8%
8	Uglašen ₅ ¹⁰⁺	1973	48	48	200	41%	2062	12%
9	Uglašen ₆ ¹⁰⁺	1959	49	49	200	39%	2062	10%
10	Uglašen ₇ ¹⁰⁺	1954	49	49	200	38%	2062	9%
11	Uglašen ₈ ¹⁰⁺	1953	49	49	200	37%	2062	10%
12	Uglašen ₉ ¹⁰⁺	1946	51	51	200	37%	2062	6%
13	Uglašen ₁₀ ¹⁰⁺	1946	51	51	200	37%	2062	8%
14	Uglašen ₁₁ ¹⁰⁺	1938	49	49	200	36%	2062	11%
15	Uglašen ₁₂ ¹⁰⁺	1924	51	51	200	35%	2062	6%
16	Uglašen ₁₃ ¹⁰⁺	1921	50	50	200	35%	2062	11%
17	Uglašen ₁₄ ¹⁰⁺	1917	50	50	200	35%	2062	13%
18	Uglašen ₁₅ ¹⁰⁺	1911	51	51	200	34%	2062	3%
19	Uglašen ₁₆ ¹⁰⁺	1904	49	49	200	32%	2062	15%
20	Uglašen ₁₇ ¹⁰⁺	1901	51	51	200	33%	2062	8%
21	Uglašen ₁₈ ¹⁰⁺	1866	52	52	200	30%	2062	7%
22	Uglašen ₁₉ ¹⁰⁺	1866	52	52	200	29%	2062	14%
23	Uglašen ₂₀ ¹⁰⁺	1844	53	53	199	28%	2060	9%
24	Rybka ₁₇₀₀	1754	21	21	1050	30%	1917	9%
25	Začetni ¹⁰⁺	1509	83	83	200	7%	2062	3%

(h) 8. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	26	26	1049	87%	1948	8%
2	Rybka ₂₁₀₀	2166	21	21	1050	75%	1948	12%
3	Uglašen ₁ ¹⁰⁺	2069	47	47	200	53%	2038	14%
4	Uglašen ₂ ¹⁰⁺	2026	48	48	200	48%	2038	12%
5	Uglašen ₃ ¹⁰⁺	2020	48	48	200	48%	2038	14%
6	Uglašen ₄ ¹⁰⁺	2019	46	46	200	47%	2038	19%
7	Uglašen ₅ ¹⁰⁺	2017	49	49	200	47%	2038	6%
8	Uglašen ₆ ¹⁰⁺	2002	49	49	200	47%	2038	9%
9	Uglašen ₇ ¹⁰⁺	1991	48	48	200	46%	2038	13%
10	Uglašen ₈ ¹⁰⁺	1988	49	49	200	45%	2038	13%
11	Uglašen ₉ ¹⁰⁺	1987	48	48	200	44%	2038	10%
12	Uglašen ₁₀ ¹⁰⁺	1976	49	49	200	42%	2038	14%
13	Uglašen ₁₁ ¹⁰⁺	1972	49	49	200	43%	2038	14%
14	Uglašen ₁₂ ¹⁰⁺	1957	49	49	200	41%	2038	10%
15	Uglašen ₁₃ ¹⁰⁺	1955	47	47	200	41%	2038	20%
16	Rybka ₁₉₀₀	1952	19	19	1050	49%	1948	14%
17	Uglašen ₁₄ ¹⁰⁺	1949	51	51	200	41%	2038	8%
18	Uglašen ₁₅ ¹⁰⁺	1944	48	48	200	41%	2038	16%
19	Uglašen ₁₆ ¹⁰⁺	1941	48	48	200	40%	2038	16%
20	Uglašen ₁₇ ¹⁰⁺	1938	51	51	200	39%	2038	6%
21	Uglašen ₁₈ ¹⁰⁺	1932	49	49	199	39%	2037	13%
22	Uglašen ₁₉ ¹⁰⁺	1902	51	51	200	36%	2038	6%
23	Uglašen ₂₀ ¹⁰⁺	1886	51	51	200	34%	2038	7%
24	Rybka ₁₇₀₀	1735	22	22	1050	24%	1948	11%
25	Začetni ¹⁰⁺	1428	91	91	200	6%	2038	2%

(i) 9. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	27	27	1049	88%	1930	7%
2	Rybka ₂₁₀₀	2184	22	22	1050	79%	1930	11%
3	Uglašen ₁ ¹⁰⁺	2068	47	47	200	51%	2047	15%
4	Uglašen ₂ ¹⁰⁺	2039	49	49	200	49%	2047	13%
5	Uglašen ₃ ¹⁰⁺	2023	48	48	200	48%	2047	14%
6	Uglašen ₄ ¹⁰⁺	1993	49	49	200	44%	2047	7%
7	Uglašen ₅ ¹⁰⁺	1987	48	48	200	41%	2047	18%
8	Uglašen ₆ ¹⁰⁺	1978	48	48	200	42%	2047	11%
9	Uglašen ₇ ¹⁰⁺	1963	49	49	200	41%	2047	9%
10	Uglašen ₈ ¹⁰⁺	1956	50	50	200	41%	2047	8%
11	Uglašen ₉ ¹⁰⁺	1956	49	49	200	41%	2047	7%
12	Rybka ₁₉₀₀	1956	19	19	1050	52%	1930	14%
13	Uglašen ₁₀ ¹⁰⁺	1953	48	48	200	40%	2047	12%
14	Uglašen ₁₁ ¹⁰⁺	1951	49	49	200	40%	2047	13%
15	Uglašen ₁₂ ¹⁰⁺	1946	50	50	200	40%	2047	12%
16	Uglašen ₁₃ ¹⁰⁺	1940	49	49	199	38%	2046	8%
17	Uglašen ₁₄ ¹⁰⁺	1939	51	51	200	37%	2047	8%
18	Uglašen ₁₅ ¹⁰⁺	1930	49	49	200	39%	2047	19%
19	Uglašen ₁₆ ¹⁰⁺	1907	51	51	200	35%	2047	8%
20	Uglašen ₁₇ ¹⁰⁺	1901	51	51	200	35%	2047	8%
21	Uglašen ₁₈ ¹⁰⁺	1888	50	50	200	34%	2047	12%
22	Uglašen ₁₉ ¹⁰⁺	1887	52	52	200	33%	2047	8%
23	Uglašen ₂₀ ¹⁰⁺	1886	51	51	200	33%	2047	10%
24	Rybka ₁₇₀₀	1750	21	21	1050	28%	1930	9%
25	Začetni ¹⁰⁺	1444	91	91	200	7%	2047	0%

(j) 10. zagon.

Rang	Ime	Rating	+	-	Igre	Rezultat	Nasp.	Remi
1	Rybka ₂₃₀₀	2300	26	26	1050	87%	1945	8%
2	Rybka ₂₁₀₀	2172	22	22	1050	76%	1945	10%
3	Uglašen ₁ ¹⁰⁺	2031	47	47	200	47%	2062	15%
4	Uglašen ₂ ¹⁰⁺	2028	48	48	200	46%	2062	8%
5	Uglašen ₃ ¹⁰⁺	2022	47	47	200	45%	2062	12%
6	Rybka ₁₉₀₀	2015	19	19	1050	58%	1945	12%
7	Uglašen ₄ ¹⁰⁺	2009	48	48	200	43%	2062	10%
8	Uglašen ₅ ¹⁰⁺	2003	49	49	200	43%	2062	8%
9	Uglašen ₆ ¹⁰⁺	1999	47	47	200	42%	2062	16%
10	Uglašen ₇ ¹⁰⁺	1992	49	49	200	41%	2062	6%
11	Uglašen ₈ ¹⁰⁺	1990	49	49	200	41%	2062	12%
12	Uglašen ₉ ¹⁰⁺	1988	48	48	200	42%	2062	13%
13	Uglašen ₁₀ ¹⁰⁺	1987	48	48	200	41%	2062	11%
14	Uglašen ₁₁ ¹⁰⁺	1971	49	49	200	40%	2062	7%
15	Uglašen ₁₂ ¹⁰⁺	1960	49	49	200	38%	2062	9%
16	Uglašen ₁₃ ¹⁰⁺	1959	49	49	200	39%	2062	12%
17	Uglašen ₁₄ ¹⁰⁺	1933	49	49	200	35%	2062	9%
18	Uglašen ₁₅ ¹⁰⁺	1927	51	51	200	35%	2062	8%
19	Uglašen ₁₆ ¹⁰⁺	1923	50	50	200	35%	2062	9%
20	Uglašen ₁₇ ¹⁰⁺	1903	48	48	200	33%	2062	19%
21	Uglašen ₁₈ ¹⁰⁺	1896	51	51	200	33%	2062	10%
22	Uglašen ₁₉ ¹⁰⁺	1896	50	50	200	31%	2062	10%
23	Uglašen ₂₀ ¹⁰⁺	1871	51	51	200	30%	2062	10%
24	Rybka ₁₇₀₀	1761	21	21	1050	28%	1945	9%
25	Začetni ¹⁰⁺	1565	75	75	200	10%	2062	2%

Osebna bibliografija

1.01 Izvirni znanstveni članek

1. BOŠKOVIĆ, Borko, BREST, Janez, ŽUMER, Viljem. Objektno orientirano načrtovanje in implementacija računalniškega šaha. *Elektroteh. vestn.*, 2006, letn. 73, no. 1, str. 31-37. [COBISS.SI-ID 10412822]
2. BREST, Janez, GREINER, Sašo, BOŠKOVIĆ, Borko, MERNIK, Marjan, ŽUMER, Viljem. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE trans. evol. comput.*. [Print ed.], dec. 2006, vol. 10, no. 6, str. 646-657. [COBISS.SI-ID 10376982]
3. BREST, Janez, BOŠKOVIĆ, Borko, GREINER, Sašo, ŽUMER, Viljem, SEPESY MAUČEC, Mirjam. Performance comparison of self-adaptive and adaptive differential evolution algorithms. *Soft computing*. [Tiskana izd.], 2007, vol. 11, no. 7, str. 617-629. <http://dx.doi.org/10.1007/s00500-006-0124-0>. [COBISS.SI-ID 11150358]
4. ZAMUDA, Aleš, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Študija samoprilagajanja krmilnih parametrov pri algoritmu DEMOwSA. *Elektroteh. vestn.*, 2008, letn. 75, št. 4, str. 223-228. [COBISS.SI-ID 12933654]
5. BREST, Janez, ZAMUDA, Aleš, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Globalna optimizacija problemov z velikim številom dimenzij. *Elektroteh. vestn.*, 2008, letn. 75, št. 5, str. 299-304. [COBISS.SI-ID 13038614]

6. BOŠKOVIĆ, Borko, BREST, Janez, ZAMUDA, Aleš, GREINER, Sašo, ŽUMER, Viljem. History mechanism supported differential evolution for chess evaluation function tuning. *Soft computing*. [Tiskana izd.], Published online: 20 March 2010. [COBISS.SI-ID 13985046]

1.05 Poljudni članek

7. BOŠKOVIĆ, Borko, BREST, Janez. Računalniški šah. *Abakus (Maribor)*, nov.-dec. 2006, let. 6, št. 5, str. 18. [COBISS.SI-ID 10908694]

1.08 Objavljeni znanstveni prispevek na konferenci

8. BOŠKOVIĆ, Borko, BREST, Janez, REBERNAK, Damijan, ŽUMER, Viljem. Načrtovanje in gradnja informacijskega sistema s trinivojsko arhitekturo. V: ZAJC, Baldomir (ur.). *Zbornik devete Elektrotehniške in računalniške konference ERK 2000, 21. - 23. september 2000, Portorož, Slovenija*. Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2000, zv. B, str. 47-50. [COBISS.SI-ID 5802262]

9. BOŠKOVIĆ, Borko, REBERNAK, Damijan, BREST, Janez, MERNIK, Marjan, ŽUMER, Viljem. Orodje za izdelavo aplikacij v spletnem informacijskem sistemu. V: ZAJC, Baldomir (ur.). *Zbornik desete Elektrotehniške in računalniške konference ERK 2001, 24. - 26. september 2001, Portorož, Slovenija*. Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2001, zv. B, str. 115-118. [COBISS.SI-ID 6542870]

10. BREST, Janez, ROŠKAR, Silvo, BOŠKOVIĆ, Borko, REBERNAK, Damijan, ŽUMER, Viljem. Algoritma za izračun sumarnih kosovnic v strojogradnji. V: ZAJC, Baldomir (ur.). *Zbornik desete Elektrotehniške in računalniške konference ERK 2001, 24. - 26. september 2001, Portorož, Slovenija*. Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2001, zv. B, str. 127-130. [COBISS.SI-ID 6543638]

11. BOŠKOVIĆ, Borko, REBERNAK, Damijan, BREST, Janez, ŽUMER, Viljem. Preslikava E-R modela v objektno orientiran model. V: ZAJC, Baldomir (ur.). *Zbornik enajste mednarodne Elektrotehniške in računalniške konference ERK 2002, 23.-25. september 2002, Portorož, Slovenija*. Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, [2002], zv. B, str. 63-66. [COBISS.SI-ID 7416854]

12. REBERNAK, Damijan, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Načrtovanje in implementacija spletnega informacijskega sistema. V: ZAJC, Baldomir (ur.). *Zbornik enajste mednarodne Elektrotehniške in računalniške konference ERK 2002, 23.-25. september 2002, Portorož, Slovenija*. Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, [2002], zv. B, str. 159-162. [COBISS.SI-ID 7426326]

13. BREST, Janez, ROŠKAR, Silvo, BOŠKOVIĆ, Borko, REBERNAK, Damijan, ŽUMER, Viljem. O pripadnosti materialov in polizdelkov v kosovni proizvodnji. V: ZAJC, Baldomir (ur.). *Zbornik enajste mednarodne Elektrotehniške in računalniške konference ERK 2002, 23.-25. september 2002, Portorož, Slovenija*. Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, [2002], zv. B, str. 187-190. [COBISS.SI-ID 7427606]

14. GREINER, Sašo, BOŠKOVIĆ, Borko, BREST, Janez, ŽUMER, Viljem. Security issues in information systems based on open-source technologies. V: ZAJC, Baldomir (ur.), TKALČIČ, Marko (ur.). *The IEEE Region 8 EUROCON 2003 : computer as a tool : 22-24. September 2003, Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia : proceedings*. Piscataway: IEEE, cop. 2003, vol. B, str. 449-453. [COBISS.SI-ID 8241174]

15. GREINER, Sašo, REBERNAK, Damijan, BOŠKOVIĆ, Borko, BREST, Janez, ŽUMER, Viljem. Web information system based on open source technologies. V: BUDIN, Leo (ur.), LUŽAR - STIFFLER, Vesna (ur.), BEKIĆ, Zoran (ur.), HLJUŽ DOBRIĆ, Vesna (ur.). *25th International Conference on Information Technology Interfaces, June 16-19, 2003, Cavtat, Croatia. ITI 2003 : proceedings of the 25th International Conference on Information Technology Interfaces, June 16-19, 2003, Cavtat,*

Croatia. Zagreb: University of Zagreb, SRCE University Computing Centre, 2003, str. 137-142. [COBISS.SI-ID 7991318]

16. BREST, Janez, ROŠKAR, Silvo, BOŠKOVIĆ, Borko, REBERNAK, Damijan, GREINER, Sašo, KRUŠEC, Robert, ŽUMER, Viljem. Informacijski sistem v proizvodnem procesu. V: ZAJC, Baldomir (ur.). *Zbornik dvanajste mednarodne Elektrotehniške in računalniške konference ERK 2003, 25. - 26. september 2003, Ljubljana, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2003, str. 365-368. [COBISS.SI-ID 8229910]

17. BOŠKOVIĆ, Borko, REBERNAK, Damijan, GREINER, Sašo, BREST, Janez, ŽUMER, Viljem. Nadzorovanje virov računalniškega sistema v operacijskem sistemu Linux. V: ZAJC, Baldomir (ur.). *Zbornik dvanajste mednarodne Elektrotehniške in računalniške konference ERK 2003, 25. - 26. september 2003, Ljubljana, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ..., 1581-4572). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2003, str. 397-400. [COBISS.SI-ID 8233494]

18. BOŠKOVIĆ, Borko, GREINER, Sašo, BREST, Janez, ŽUMER, Viljem. The representation of chess game. V: LUŽAR - STIFFLER, Vesna (ur.), HLJUŽ DOBRIĆ, Vesna (ur.). *27th International Conference on Information Technology Interfaces, June 20-23, 2005, Cavtat, Croatia. ITI 2005 : proceedings of the 27th International Conference on Information Technology Interfaces, June 20-23, 2005, Cavtat, Croatia*, (IEEE Catalog, No. 05EX1001). Zagreb: University of Zagreb, SRCE University Computing Centre, [2005], str. 381-386. [COBISS.SI-ID 9658134]

19. BREST, Janez, GREINER, Sašo, BOŠKOVIĆ, Borko, ŽUMER, Viljem. A heuristic algorithm for function optimization. V: BUDIN, Leo (ur.), RIBARIĆ, Slobodan (ur.). *MIPRO 2005 : 28. međunarodni skup, May/Svibanj 30 - June/Lipanj 03, 2005, Opatija, Croatia : Proceedings/Zbornik radova*. Rijeka: MIPRO, 2005, str. 91-94.

[COBISS.SI-ID 9590550]

20. BOŠKOVIĆ, Borko, GREINER, Sašo, BREST, Janez, ŽUMER, Viljem. Učenje računalniškega šaha z uporabo algoritma diferencialne evolucije. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik štirinajste mednarodne Elektrotehniške in računalniške konference ERK 2005, 26. - 28. september 2005, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2005, zv. B, str. 71-74. [COBISS.SI-ID 9867798]

21. BREST, Janez, BOŠKOVIĆ, Borko, GREINER, Sašo, ŽUMER, Viljem. Nastavitev parametrov pri algoritmu diferencialne evolucije. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik štirinajste mednarodne Elektrotehniške in računalniške konference ERK 2005, 26. - 28. september 2005, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2005, zv. B, str. 79-82. [COBISS.SI-ID 9868054]

22. BOŠKOVIĆ, Borko, GREINER, Sašo, BREST, Janez, ŽUMER, Viljem. A differential evolution for the tuning of a chess evaluation function. V: 2006 IEEE World Congress on Computational Intelligence, Vancouver, Canada, July 16-21, 2006. *2006 IEEE World Congress on Computational Intelligence : Vancouver, BC, Canada, July 16-21, 2006 : a joint conference of the: 2006 International Conference on Neural Networks, 2006 IEEE International Conference on Fuzzy Systems, 2006 IEEE Congress on Evolutionary Computation*. [Piscataway]: The Institute of Electrical and Electronics Engineering: = IEEE, cop. 2006, str. 6742-6747. [COBISS.SI-ID 10657046]

23. BOŠKOVIĆ, Borko, GREINER, Sašo, BREST, Janez, ŽUMER, Viljem. Adaptivni algoritem diferencialne evolucije za ugaševanje parametrov ocenitve funkcije računalniškega šaha. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik petnajste mednarodne Elektrotehniške in računalniške konference ERK 2006, 25. - 27. september 2006, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference

ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2006, zv. B, str. 83-86. [COBISS.SI-ID 10768150]

24. BREST, Janez, SEPESY MAUČEC, Mirjam, BOŠKOVIĆ, Borko, GREINER, Sašo, ŽUMER, Viljem. Optimizacija z omejitvami: eksperimentalni rezultati s samoprilagodljivim algoritmom diferencialne evolucije. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik petnajste mednarodne Elektrotehniške in računalniške konference ERK 2006, 25. - 27. september 2006, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2006, zv. B, str. 91-94. [COBISS.SI-ID 10768406]

25. ZAMUDA, Aleš, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Differential evolution for multiobjective optimization with self adaptation. V: *CEC 2007 : 2007 Congress on Evolutionary Computation : 25-28 September 2007, Singapore*. Piscataway: Institute of Electrical and Electronics Engineers, 2007, str. 3617-3624. [COBISS.SI-ID 11730710]

26. BOŠKOVIĆ, Borko, BREST, Janez, ZAMUDA, Aleš, ŽUMER, Viljem. Uglasavanje šahovskega programa BBChess z uporabo algoritma diferencialne evolucije. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, 24. - 26. september 2007, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2007, zv. B, str. 73-76. [COBISS.SI-ID 11760662]

27. BREST, Janez, ZAMUDA, Aleš, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Večkriterijska optimizacija: primerjava algoritmov v MOjDE in DEMO. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, 24. - 26. september 2007, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8,

Slovenska sekcija IEEE, 2007, zv. B, str. 85-88. [COBISS.SI-ID 11773718]

28. ZAMUDA, Aleš, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Večkriterijska optimizacija: eksperimentalni rezultati algoritmov MOjDE in DEMO. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, 24. - 26. september 2007, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2007, zv. B, str. 89-92. [COBISS.SI-ID 11773974]

29. BREST, Janez, ZAMUDA, Aleš, BOŠKOVIĆ, Borko, GREINER, Sašo, SEPESY MAUČEC, Mirjam, ŽUMER, Viljem. Self-adaptive differential evolution with SQP local search. V: FILIPIČ, Bogdan (ur.), ŠILC, Jurij (ur.). *Bioinspired optimization methods and their applications : proceedings of the Third International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2008, 13-14 October 2008, Ljubljana, Slovenia*. Ljubljana: Jožef Stefan Institute, 2008, str. 59-69. [COBISS.SI-ID 12707862]

30. BREST, Janez, ZAMUDA, Aleš, BOŠKOVIĆ, Borko, SEPESY MAUČEC, Mirjam, ŽUMER, Viljem. High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction. V: IEEE World Congress on Computational Intelligence, Hong Kong, June 1-6, 2008. *WCCI 2008 Proceedings*. [S. l.]: IEEE, 2008, str. 2032-2039. [COBISS.SI-ID 12328982]

31. ZAMUDA, Aleš, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution. V: IEEE World Congress on Computational Intelligence, Hong Kong, June 1-6, 2008. *WCCI 2008 Proceedings*. [S. l.]: IEEE, 2008, str. 3719-3726. [COBISS.SI-ID 12329238]

- 32.** ZAMUDA, Aleš, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Diferencialna evolucija za večkriterijsko optimizacijo s samoprilagajanjem in z lokalnim preiskovanjem SQP. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik sedemnajste mednarodne Elektrotehniške in računalniške konference ERK 2008, 29. september - 1. oktober 2008, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2008, zv. B, str. 103-106. [COBISS.SI-ID 12695574]
- 33.** BOŠKOVIĆ, Borko, BREST, Janez, ZAMUDA, Aleš, ŽUMER, Viljem. Ratiniranje pri uglasovanju šahovskega programa z algoritmom diferencialne evolucije. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik sedemnajste mednarodne Elektrotehniške in računalniške konference ERK 2008, 29. september - 1. oktober 2008, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2008, zv. B, str. 123-126. [COBISS.SI-ID 12695830]
- 34.** ZAMUDA, Aleš, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Differential evolution with self-adaptation and local search for constrained multiobjective optimization. V: *2009 IEEE Congress on Evolutionary Computation, IEEE CEC 2009, 18th to 21st May 2009, Trondheim, Norway : programme & abstracts*. Piscataway: IEEE, 2009, str. 195-202. [COBISS.SI-ID 13217558]
- 35.** BREST, Janez, ZAMUDA, Aleš, BOŠKOVIĆ, Borko, SEPESY MAUČEC, Mirjam, ŽUMER, Viljem. Dynamic optimization using self-adaptive differential evolution. V: *2009 IEEE Congress on Evolutionary Computation, IEEE CEC 2009, 18th to 21st May 2009, Trondheim, Norway : programme & abstracts*. Piscataway: IEEE, 2009, str. 415-422. [COBISS.SI-ID 13217814]
- 36.** PULKO, Milan, BOŠKOVIĆ, Borko, BREST, Janez. Programiranje grafičnih vmesnikov s knjižnico QT. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik*

Osemnajste mednarodne elektrotehniške in računalniške konference - ERK 2009, 21-23. september 2009, Portorož, Slovenija. Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2009, zv. B, str. 27-30, ilustr. [COBISS.SI-ID 13463062]

1.12 Objavljeni povzetek znanstvenega prispevka na konferenci

37. BOŠKOVIĆ, Borko, BREST, Janez. Objektno orientirano načrtovanje in implementacija računalniškega šaha. V: ZAJC, Baldomir (ur.). *Zbornik trinajste mednarodne elektrotehniške in računalniške konference ERK 2004, 27. - 29. september 2004, Portorož, Slovenija.* Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, [2004?], zv. B, str. 299-300. [COBISS.SI-ID 9111574]

38. BOŠKOVIĆ, Borko. Predstavitev članka Learning with case-injected genetic algorithms. V: FILIPIČ, Bogdan (ur.), MERNIK, Marjan (ur.), PAPA, Gregor (ur.). *Zbornik povzetkov delavnic "Algoritmi po vzorih iz narave" v študijskem letu 2004/2005 : Ljubljana, september 2005.* Ljubljana: Institut "Jožef Stefan", 2005, str. 11. [COBISS.SI-ID 12780822]

39. BOŠKOVIĆ, Borko. Učenje računalniškega šaha z evolucijskim pristopom. V: FILIPIČ, Bogdan (ur.), MERNIK, Marjan (ur.), PAPA, Gregor (ur.). *Zbornik povzetkov delavnic "Algoritmi po vzorih iz narave" v študijskem letu 2004/2005 : Ljubljana, september 2005.* Ljubljana: Institut "Jožef Stefan", 2005, str. 16. [COBISS.SI-ID 12781078]

40. BOŠKOVIĆ, Borko. Učenje računalniškega šaha z uporabo algoritma diferencialne evolucije. V: FILIPIČ, Bogdan (ur.), MERNIK, Marjan (ur.), PAPA, Gregor (ur.). *Zbornik povzetkov delavnic "Algoritmi po vzorih iz narave" v študijskem letu 2004/2005 : Ljubljana, september 2005.* Ljubljana: Institut "Jožef Stefan", 2005, str. 27. [COBISS.SI-ID 12781846]

41. BOŠKOVIĆ, Borko. Adaptivni algoritem diferencialne evolucije za uglaševanje parametrov ocenitve funkcije računalniškega šaha. V: FILIPIČ, Bogdan (ur.), MERNIK, Marjan (ur.), PAPA, Gregor (ur.). *Zbornik povzetkov delavnic "Algoritmi po vzorih iz narave"v študijskem letu 2005/2006 : Ljubljana, september 2006*. Ljubljana: Institut "Jožef Stefan", 2006, str. 24. [COBISS.SI-ID 12777750]

42. BOŠKOVIĆ, Borko. AODE - al[o]goritem diferencialne evolucije, ki temelji na nasprotjih. V: FILIPIČ, Bogdan (ur.), MERNIK, Marjan (ur.), PAPA, Gregor (ur.). *Zbornik povzetkov delavnic "Algoritmi po vzorih iz narave"v študijskem letu 2006/2007 : Ljubljana, september 2007*. Ljubljana: Institut "Jožef Stefan", 2007, str. 10. [COBISS.SI-ID 12762902]

43. BOŠKOVIĆ, Borko. Uglaševanje šahovskega programa BBChess z uporabo algoritma diferencialne evolucije. V: FILIPIČ, Bogdan (ur.), MERNIK, Marjan (ur.), PAPA, Gregor (ur.). *Zbornik povzetkov delavnic "Algoritmi po vzorih iz narave"v študijskem letu 2006/2007 : Ljubljana, september 2007*. Ljubljana: Institut "Jožef Stefan", 2007, str. 24. [COBISS.SI-ID 12767766]

44. BOŠKOVIĆ, Borko. Ratingiranje pri uglaševanju šahovskega programa z algoritmom diferencialne evolucije. V: FILIPIČ, Bogdan (ur.), MERNIK, Marjan (ur.), PAPA, Gregor (ur.). *Zbornik povzetkov delavnic "Algoritmi po vzorih iz narave"v študijskem letu 2007/2008 : Ljubljana, september 2008*. Ljubljana: Institut "Jožef Stefan", 2008, str. 26. [COBISS.SI-ID 12756758]

1.13 Objavljeni povzetek strokovnega prispevka na konferenci

45. BREST, Janez, REBERNAK, Damijan, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Uporaba Linuxa pri gradnji informacijskega sistema v podjetju = Enterprise information system with the use of Linux. V: 1. mednarodni festival KIBLIX 2002 - IT Linux festival, 7., 8., in 9. november 2002, Maribor, Slovenija. *Katalog*. Maribor: Multimedia center Kibla, 2002, str. 34-36. [COBISS.SI-ID 7513110]

1.16 Samostojni znanstveni sestavek ali poglavje v monografski publikaciji

46. BREST, Janez, ZAMUDA, Aleš, BOŠKOVIĆ, Borko, GREINER, Sašo, ŽUMER, Viljem. An analysis of the control parameters' adaptation in DE. V: *Advances in differential evolution*, (Studies in computational intelligence, Vol. 143). Berlin; Heidelberg: Springer, 2008, str. 89-110. [COBISS.SI-ID 12503830]

47. BOŠKOVIĆ, Borko, GREINER, Sašo, BREST, Janez, ZAMUDA, Aleš, ŽUMER, Viljem. An adaptive differential evolution algorithm with opposition-based mechanisms, applied to the tuning of a chess program. V: *Advances in differential evolution*, (Studies in computational intelligence, Vol. 143). Berlin; Heidelberg: Springer, 2008, str. 287-298. [COBISS.SI-ID 12504854]

MONOGRAFIJE IN DRUGA ZAKLJUČENA DELA

2.05 Drugo učno gradivo

48. ŽUMER, Viljem, MERNIK, Marjan, BREST, Janez, KRUŠEC, Robert, AVDIČAUŠEVIĆ, Enis, REBERNAK, Damijan, BOŠKOVIĆ, Borko. *Programski jezik C : gradivo za tečaj: nadgradnja osnov C-N-11-01*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, Laboratorij za računalniške arhitekture in jezike, 2001. 1 zv. (loč. pag.). [COBISS.SI-ID 6794006]

49. ŽUMER, Viljem, MERNIK, Marjan, BREST, Janez, KRUŠEC, Robert, LENIČ, Mitja, AVDIČAUŠEVIĆ, Enis, REBERNAK, Damijan, BOŠKOVIĆ, Borko. *Programski jezik C++ : gradivo za tečaj*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, Laboratorij za računalniške arhitekture in jezike, 2001. 218 f. [COBISS.SI-ID 6054934]

50. ŽUMER, Viljem, MERNIK, Marjan, BREST, Janez, KRUŠEC, Robert, ČREPINŠEK, Matej, KOSAR, Tomaž, GREINER, Sašo, REBERNAK, Damijan, BOŠKOVIĆ,

Borko, TUTEK, Simon. *Operacijski sistem LINUX/UNIX : gradivo za nadaljevalni tečaj*, (Računalniško opismenjevanje MŠZŠ). Maribor: Laboratorij za računalniške arhitekture in jezike in Center za programske jezike, 2003. 48 f. [COBISS.SI-ID 8160790]

51. ŽUMER, Viljem, MERNIK, Marjan, BREST, Janez, KRUŠEC, Robert, ČREPINŠEK, Matej, KOSAR, Tomaž, GREINER, Sašo, REBERNAK, Damijan, BOŠKOVIĆ, Borko. *Operacijski sistem LINUX/UNIX : gradivo za začetni tečaj*, (Računalniško opismenjevanje MŠZŠ). Maribor: Laboratorij za računalniške arhitekture in jezike in Center za programske jezike, 2003. 27 f. [COBISS.SI-ID 8160534]

2.11 Diplomsko delo

52. BOŠKOVIĆ, Borko. *Orodje za izdelavo aplikacij v spletnem informacijskem sistemu : diplomsko delo visokošolskega strokovnega študija*, (Fakulteta za elektrotehniko, računalništvo in informatiko, Diplomska dela visokošolskega strokovnega študija). [Maribor]: [B. Boškovič], 2002. XIII, 59 f., ilustr. [COBISS.SI-ID 6947094]

53. BOŠKOVIĆ, Borko. *Implementacija računalniškega šaha : diplomsko delo univerzitetnega študija*, (Fakulteta za elektrotehniko, računalništvo in informatiko, Diplomska dela univerzitetnega študija). [Maribor]: [B. Boškovič], 2004. XI, 65 f., ilustr. [COBISS.SI-ID 8809238]

2.12 Končno poročilo o rezultatih raziskav

54. ŽUMER, Viljem, BREST, Janez, REBERNAK, Damijan, BOŠKOVIĆ, Borko. *Programska oprema za nadzor in upravljanje procesa proizvodnje : poročilo za leto 2001 : projekt za podjetje Lestro Ledinek Engineering d.o.o.* Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, 2002. 33 f. [COBISS.SI-ID 6909718]

55. BREST, Janez, GREINER, Sašo, REBERNAK, Damijan, BOŠKOVIĆ, Borko, TUTEK, Simon, ŽUMER, Viljem. *Informacijski sistem za projektno vodenje proizvodnje v strojogradnji : končno poročilo*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, 2005. 8 f., [25] f. [COBISS.SI-ID 9460502]

56. BREST, Janez, BOŠKOVIĆ, Borko, GREINER, Sašo, REBERNAK, Damijan, ŽUMER, Viljem. *Izdelava programske opreme za nadzor in upravljanje dela procesa proizvodnje : končno poročilo*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, 2005. 5 f. [COBISS.SI-ID 10210838]

2.21 Programska oprema

57. BOŠKOVIĆ, Borko, BREST, Janez. *Programska oprema JCHESS*. Maribor: Institut za računalništvo, 2005. 1 CD-ROM. [COBISS.SI-ID 10259478]

58. ZAMUDA, Aleš, STRNAD, Damjan, BREST, Janez, BOŠKOVIĆ, Borko. *Drevo*. [s. l.: s. n., 2007]. 1 CD ROM. <http://labraj.uni-mb.si/~ales/codes/drevo.zip>. [COBISS.SI-ID 11607062]

59. BOŠKOVIĆ, Borko, BREST, Janez, ZAMUDA, Aleš. *Šahovski program BBChess 1.1*. Maribor: [s. n., 2007]. 1 CD ROM. http://labraj.uni-mb.si/images/6/6e/BBChess1.1_src.tar.bz. [COBISS.SI-ID 11873046]

2.25 Druge monografije in druga zaključena dela

60. ŽUMER, Viljem, MERNIK, Marjan, BREST, Janez, KRUŠEC, Robert, LENIČ, Mitja, AVDIČAUŠEVIČ, Enis, REBERNAK, Damijan, BOŠKOVIĆ, Borko, VREŽE, Aleksander. *Operacijski sistem LINUX-UNIX : nadaljevalni*, (Računalniško opismenjevanje). Ljubljana: Ministrstvo za šolstvo in šport: Zavod Republike Slovenije za

šolstvo, 2000. 72 f. [COBISS.SI-ID 5400342]

61. ŽUMER, Viljem, MERNIK, Marjan, BREST, Janez, KRUŠEC, Robert, AVDIČAUŠEVIĆ, Enis, REBERNAK, Damijan, BOŠKOVIĆ, Borko. *Operacijski sistem LINUX/UNIX : gradivo za nadaljevalni tečaj*, (Računalniško opismenjevanje). Maribor: Ministrstvo za šolstvo in šport: Zavod Republike Slovenije za šolstvo, 2001. 81 f. [COBISS.SI-ID 6716694]

62. ŽUMER, Viljem, BREST, Janez, MERNIK, Marjan, ČREPINŠEK, Matej, KRUŠEC, Robert, KOSAR, Tomaž, REBERNAK, Damijan, BOŠKOVIĆ, Borko, TUTEK, Simon, GERLIČ, Goran, GREINER, Sašo. *LINUX in Qt : tečaj za AMES, Brezovica pri Ljubljani*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, Laboratorij za računalniške arhitekture in jezike, 2003. 1 mapa (loč. pag.). [COBISS.SI-ID 7801110]

IZVEDENA DELA (DOGODKI)

3.25 Druga izvedena dela

63. BOŠKOVIĆ, Borko, GREINER, Sašo, ZAMUDA, Aleš. *Tečaj Linux : tečaj za Dravske elektrarne Maribor (DEM) in Holding slovenskih elektrarn (HSE), Maribor, 4.-12. junij 2009*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, Laboratorij za računalniške arhitekture in jezike, 2009. [COBISS.SI-ID 13264662]

64. BOŠKOVIĆ, Borko. *Šahovski program : razvojno-raziskovalni prispevek Laboratorija za računalniške arhitekture in jezike na Delavnici programske skupine Računalniški sistemi, metodologije in inteligentne storitve, Fakulteta za elektrotehniko, računalništvo in informatiko [Maribor], 10. februar 2010*. Maribor, 2010. [COBISS.SI-ID 13968150]

Življenjepis

Borko Bošković se je rodil v Murski Soboti dne 22.08.1977. Osnovno šolo je obiskoval v Lendavi od leta 1984. Po končani osnovni šoli se je leta 1992 vpisal v Srednjo tehnično šolo v Somboru, ki jo je leta 1996 uspešno zaključil.

Nato se je še v istem letu vpisal na Fakulteto za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, in sicer v visokošolski strokovni študijski program Računalništvo in informatika. Leta 2002 je kot drugi študent na omenjeni smeri diplomiral s povprečno oceno 8,2.

Leta 2003 je nadaljeval študij na univerzitetnem študijskem programu Računalništvo in informatika. Kot prvi v letniku je diplomiral dne 05.04.2004 s povprečno oceno 9,5. V diplomski nalogi "Implementacija računalniškega šaha" je implementiral šahovski program, ki ima moč igranja več kot 2400 rating točk oz. igra na nivoju mednarodnega mojstra in uvršča se med 20 najboljših odprto-kodnih šahovskih programov na vseh mednarodnih lestvicah.

Že leta 2000 se je zaposlil v Laboratoriju za računalniške arhitekture in jezike, ki ga vodi red. prof. dr. Viljem Žumer, na Fakulteti za elektrotehniko, računalništvo in informatiko, kot tehniški sodelavec. V letu 2004 se je v istem laboratoriju zaposlil na delovnem mestu asistenta. Borko Bošković se trenutno ukvarja z evolucijskimi algoritmi, računalniškim šahom, programskimi jeziki in operacijskim sistemom GNU Linux.

UNIVERZA V MARIBORU

Fakulteta za elektrotehniko, računalništvo in informatiko
(ime fakultete)

IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE ZAKLJUČNEGA DELA IN
OBJAVI OSEBNIH PODATKOV AVTORJA

Ime in priimek avtorja (avtorice): Borko Bošković
Vpisna številka: 95028737
Študijski program: Računalništvo in informatika
Naslov zaključnega dela: Uglaševanje šahovske ocenitvene funkcije s pomočjo
algoritma diferencialne evolucije
Mentor: izr. prof. dr. Janez Brest
Somentor: red. prof. dr. Viljem Žumer

Podpisani-a Borko Bošković izjavljam, da sem za potrebe arhiviranja oddal-a elektronsko verzijo zaključnega dela v Digitalno knjižnico Univerze v Mariboru. Zaključno delo sem izdelal-a sam-a ob pomoči mentorja. V skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah (Ur. l. RS, št. 16/2007) dovoljujem, da se zgoraj navedeno zaključno delo objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Tiskana verzija zaključnega dela je istovetna elektronski verziji, ki sem jo oddal-a za objavo v Digitalno knjižnico Univerze v Mariboru. Podpisani-a izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto in kraj rojstva, datum zagovora, naslov zaključnega dela) na spletnih straneh in v publikacijah UM.

Kraj in datum: Maribor, 25.05.2010

Podpis avtorja (avtorice): 