

Analiza algoritmov dinamične optimizacije s pasovi zaupanja na podlagi ratingov

Matej Moravec, Miha Ravber, Matej Črepinšek

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor
E-pošta: matej.moravec@um.si

Analysis of dynamic optimization algorithms with confidence bands based on ratings

Each newly proposed algorithm must be evaluated against current state-of-the-art algorithms in its respective field. In dynamic optimization, the goal is not only to find the global optimum at the end of the process but also to track its changes throughout the process. In this study, we present a novel method for comparing dynamic optimization algorithms that aim to find and track the evolving optimum of a dynamic optimization problem. Our method employs the Glicko-2 chess rating system to represent the performance of each algorithm with a rating value. The performance of the algorithms is monitored throughout the entire optimization process. We compared the three most commonly cited dynamic optimization algorithms in the literature, focusing on single-objective dynamic optimization problems. The evaluation was conducted using the widely recognized Moving Peaks benchmark, and the results are visualized graphically for clarity and straightforward interpretation. The advantage of our method is that the global optimum does not need to be known in advance. Statistically significant differences between the compared algorithms are easily recognizable in the graph.

1 Uvod

V praksi srečujemo različne optimizacijske probleme, pri čemer so nekateri zaradi svoje narave negotovi in se čez čas spreminjajo, kar zahteva uporabo optimizacijskih algoritmov, ki so se sposobni učinkovito odzivati na te spremembe [6]. Kriterijska funkcija ali omejitve se pri takih problemih spreminjajo in se v literaturi označujejo kot dinamični optimizacijski problemi (angl. Dynamic Optimization Problems - DOPs) [4]. DOP se naprej deli na enokriterijske in večkriterijske, vendar se v prispevku omejimo na prve. Cruz in Juan [4] sta definirala DOP na naslednji način:

$$DOP = \left\{ \begin{array}{l} \text{optimiziraj} \quad f(x, t) \\ \text{tako, da} \quad x \in F(t) \subseteq S, t \in T \end{array} \right\},$$

kjer $S \in R^n$, S predstavlja iskalni prostor, t je čas, $f : S \times T \rightarrow R$ je kriterijska funkcija, ki dodeli številsko vrednost ($f(x, t) \in R$) vsaki možni rešitvi ($x \in S$) v času

t in $F(t)$ je množica izvedljivih rešitev (angl. feasible solutions) $x \in F(t) \subseteq S$ v času t .

Spremembe pri DOP so neizogibne, zato je za njihovo reševanje potrebno nenehno prilagajanje. Ključno je, da algoritem učinkovito najde optimalno rešitev in ji tudi ob spremembah okolja sledi skozi čas [6]. Evolucijski algoritmi in metode inteligence rojev (angl. swarm intelligence) se zaradi njihove naravne zmožnosti obvladovanja sprememb okolja pogosto uporabljajo pri optimizaciji DOP [7]. Vendar jih neposredno ni mogoče uporabiti za reševanje DOP, saj so bile te metode prvotno zasnovane za optimizacijo v statičnih okoljih, kjer se kriterijska funkcija in omejitve tekom optimizacije ne spreminjajo. Zato se običajno uporabljajo skupaj z določenimi komponentami, ki tvorijo algoritme dinamične optimizacije (angl. Dynamic Optimization Algorithms - DOAs) [12]. V literaturi je mogoče zaslediti vedno več novih in izboljšanih DOA. Avtorji teh DOA morajo svoje delo primerjati z obstoječimi najsodobnejšimi algoritmi z uporabo primerjalnih testov (angl. benchmarking). Pri ovrednotenju kakovosti rešitev, ki jih najdejo posamezni DOA, se uporabljajo različni indikatorji kakovosti. Ti indikatorji se nato uporabijo za primerjavo kakovosti posameznih DOA. V svojem prispevku smo za primerjavo uporabili šahovski sistem rangiranja, ki z uporabo določenih enačb in pravil vpliva na oceno kakovosti igralca. Ocena kakovosti je izražena z ratingom, ki se spreminja glede na doseženo število zmag, porazov in remijev na organiziranem turnirju. Za primerjavo smo uporabili tri najbolj citirane enokriterijske DOA, ki smo jih rangirali pred vsako spremembo okolja. Za lažjo interpretacijo rezultatov smo ratinge posameznih DOA prikazali na grafu, iz katerega je razvidno, kako uspešen je posamezni DOA v primerjavi z drugimi, ki sodelujejo na turnirju. V naslednjem poglavju opišemo indikatorje kakovosti, ki se uporabljajo na področju primerjave več DOA. V 3. poglavju opišemo predlagano metodo primerjave več DOA. Rezultate eksperimenta vključno z grafi predstavimo v poglavju 4. V zadnjem, 5. poglavju povzamemo predlagano metodo primerjave in predstavimo možnosti razširitve uporabe.

2 Indikatorji kakovosti

Za ovrednotenje kakovosti rešitve, ki jo dobimo pri reševanju enokriterijskega DOP, se uporabljajo indika-

torji kakovosti. Z njimi se izračuna, kako kakovostna je rešitev, ki jo je našel algoritem, pri čemer izračunana vrednost služi za primerjavo uspešnosti le-tega. Na področju enokriterijske dinamične optimizacije obstaja več indikatorjev kakovosti, ki se delijo na tiste, ki temeljijo na uspešnosti/napaki (angl. fitness/error-based) in tiste, ki temeljijo na učinkovitosti (angl. efficiency-based). V literaturi se pogosteje uporabljajo prvi, pri katerih mora biti globalni optimum vsakega okolja znan, da se lahko izračuna napaka dobljene rešitve. Omeniti velja, da pri problemih iz resničnega sveta globalni optimum ni na voljo, medtem ko je v večini sintetičnih DOP globalni optimum znan vnaprej. Najpogosteje uporabljena indikatorja med njimi sta offline-error (E_o) in E_{BBC} [13].

Drugačen način primerjave uspešnosti posameznih algoritmov, predstavlja šahovski sistem rangiranja (angl. chess rating system). Pri taki primerjavi algoritmi predstavljajo igralce, ki se udeležijo turnirja. Kakovost vsakega igralca na turnirju je določena glede na njegovo uspešnost v igrah z ostalimi igralci. Rating predstavlja kakovost igralca in le-ta se mu poveča, če igra bolje od pričakovanega oz. zmanjša, če igra slabše od pričakovanega. Sprememba ratinga je odvisna od ratinga nasprotnika. Za primerjavo uspešnosti enokriterijskih DOA smo uporabili ogrodje EARS (angl. Evolutionary Algorithm's Rating System), ki pri primerjavi uporablja šahovski sistem rangiranja Glicko-2. [11]

2.1 Indikator E_o

Indikator E_o [3] je napogosteje uporabljen pristop v literaturi in z uporabo enačbe (1) izračuna povprečno napako najboljšega najdenega položaja v vseh ovrednotenjih kriterijske funkcije.

$$E_o = \frac{1}{Tv} \sum_{t=1}^T \sum_{c=1}^v \left(f^{(t)}(\vec{x}^{*(t)}) - f^{(t)}(\vec{x}^{*((t-1)v+c)}) \right) \quad (1)$$

Kjer je $\vec{x}^{*(t)}$ položaj globalnega optimuma v okolju t , T je število vseh okolij, v je pogostost pojavljanja sprememb, c predstavlja števec ovrednotenih kriterijske funkcije za vsako okolje, in $\vec{x}^{*((t-1)v+c)}$ je najboljši najdeni položaj c -tega ovrednotenja kriterijske funkcije v okolju t .

2.2 Indikator E_{BBC}

Drugi najpogosteje uporabljeni indikator, ki temelji na uspešnosti/napaki, je indikator E_{BBC} [10]. Indikator E_{BBC} v enačbi (2) pri izračunu upošteva le zadnjo napako, preden pride do spremembe okolja (tj. na koncu vsakega okolja).

$$E_{BBC} = \frac{1}{T} \sum_{t=1}^T \left(f^{(t)}(\vec{x}^{*(t)}) - f^{(t)}(\vec{x}^{*(t)}) \right) \quad (2)$$

Kjer je $\vec{x}^{*(t)}$ najboljši najdeni položaj v okolju t in je pridobljen na koncu vsakega okolja.

2.3 Ogrodje EARS

Ogrodje za rangiranje evolucijskih algoritmov EARS [11] se uporablja za pošteno in enostavno primerjavo evolucijskih algoritmov. Je odprtokodno ogrodje, razvito v programskem jeziku Java. Ogrodje za primerjavo evolucijskih algoritmov uporablja šahovski sistem rangiranja Glicko-2, ki glede na določene enačbe in pravila vpliva na rating igralca (algoritma). V splošnem velja, da je rating pozitivno celo število, ki predstavlja uspešnost igralca in se spreminja, ko igralec na organiziranem turnirju doseže določeno število zmag, porazov in remijev. Uspešnejši je tisti igralec, ki ima višji rating. V ogrođu EARS so algoritmi predstavljeni kot šahisti in na turnirju tekmujejo v dvojicah. Vsak pri tem odigra N -iger na istem problemu z vsemi drugimi, udeleženi na turnirju, in pri tem poskusi najti čim boljše rešitev. Na tekmovanju se med dvema algoritmoma primerjata dobljeni rešitvi vsakega algoritma in določi se izid igre, ki je lahko zmaga enega algoritma in poraz drugega ali pa je izid neodločen. Po vseh odigranih igrah, se po sistemu Glicko-2 vsakemu algoritmu izračuna rating R , odklon ratinga RD (angl. Rating Deviation) in interval ratinga RI (angl. Rating Interval). Po končanem turnirju se algoritmi razvrstijo glede na njihov rating, pri čemer je boljši tisti algoritem, ki ima višji rating. Odklon ratinga izraža zanesljivost ratinga algoritma in se izračuna glede na rating in odklon ratinga. Pri dobljenih rezultatih lahko s 95 % gotovostjo trdimo, da se rating algoritma nahaja v intervalu $[R - 2RD, R + 2RD]$. Prekrivanje intervalov ratinga (RI) določa statistični pomen. Dva algoritma sta signifikantno različna, kadar se dva intervala ne prekrivata.

3 Predlagana metoda

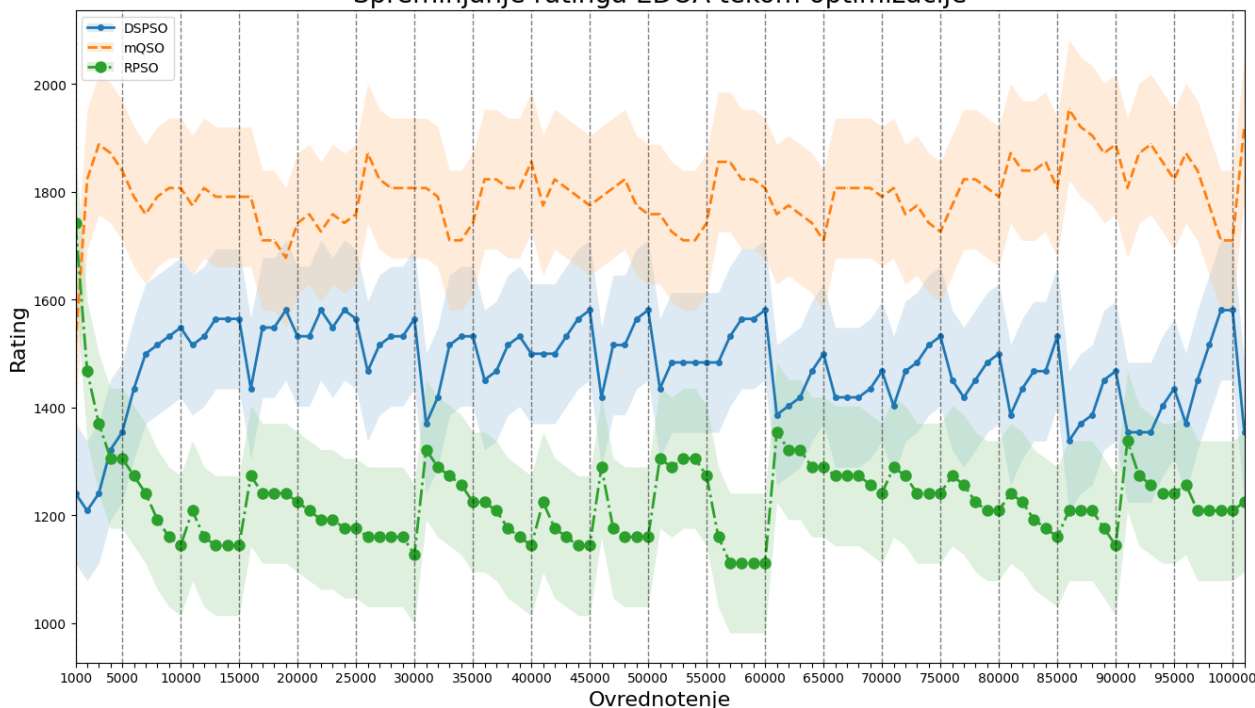
Pri primerjavi več DOA smo uporabili odprtokodno platformo EDOLAB [9], razvito v programskem jeziku MATLAB, ki v trenutni verziji vsebuje 25 DOA, 3 primerjalne teste in 2 indikatorja kakovosti. Izmed nabora vseh implementiranih DOA v platformi, smo za primerjavo izbrali multi Q-learning-based Swarm Optimization (mQSO) [1], Dynamic Species-based Particle Swarm Optimization (DSPSO) [8] in Random Particle Swarm Optimization (RPSO) [5], ki so v literaturi najbolj citirani. Vse DOA smo primerjali na standardni različici primerjalnega testa Moving Peaks benchmark (MPB) [2], ki je najbolj popularen na področju dinamične optimizacije [13].

Za enakovredno primerjavo smo pri vseh zagonih DOA uporabili isti generator naključnih števil in isto seme. Vsak DOA je reševal primerjalni test MPB z istimi parametri (tabela 1).

Tabela 1: Parametri primerjalnega testa MPB.

Parameter	Vrednost
Število vrhov	10
Pogostost sprememb	5000
Število dimenzij	5
Resnost premika	1
Število okolij	100

Spreminjanje ratinga EDOA tekom optimizacije



Slika 1: Ratingi EDOA skozi prvih 20 okolij problema.

Izbrane DOA smo primerjali na vsakih 1000 ovrednotenij tako, da smo uporabili odprtokodno ogrodje EARS, kjer je vsak algoritem z vsakim drugim algoritmom odigral 31 iger, dobil svoj rating in bil rangiran glede na šahovski sistem rangiranja Glicko-2. Pri primerjavi smo uporabili trenutno najboljšo najdeno rešitev. Tako smo na koncu primerjali algoritme 5000-krat in ratinge DOA prikazali na sliki 1. Na sliki 1 je meja med vsakim novim okoljem označena z navpično črtkano črto.

3.1 Primerjalni test MPB

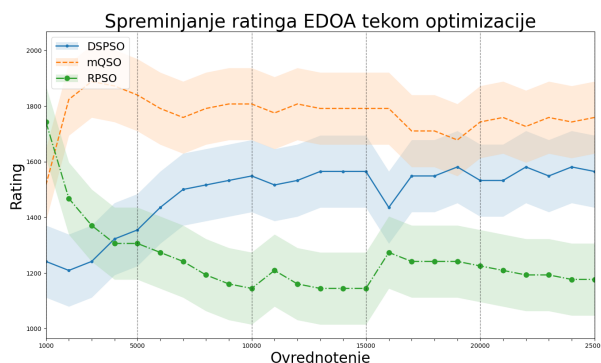
Standardna različica MPB [2] ustvari pokrajine s sestavljanjem serije pravilnih, unimodalnih, simetričnih in popolnoma neločljivih (angl. fully non-separable) stožčastih obetavnih območij (tj. vrhov). V literaturi sicer obstaja več novejših različic MPB, vendar ker je glavna ideja prispevka zgolj primerjava obnašanja algoritmov med pojavljanjem sprememb in ne kako uspešno posamezni algoritem reši dani problem, smo se odločili za uporabo v literaturi najpogosteje uporabljene standardne različice MPB.

4 Rezultati eksperimenta

V tem poglavju predstavimo rezultate predlagane metode primerjave več DOA. Peng in drugi [9] navajajo, da so za zagotavljanje poštene primerjave v eksperimentih pomenotitli nekatere vidike optimizacijskih komponent uporabljenih v DOA in da zato nekateri DOA niso natančno taki kot v izvornih publikacijah. Z namenom, da bi zagotovili pošteno primerjavo, nismo pri primerjavi mQSO, DSPSO in RPSO spreminjali implementacije nobenega DOA. Predpostavljajo, da so vsi DOA obveščeni o spremembah okolja in zato ne uporabljajo nobene kompo-

nente za zaznavo sprememb v DOA. V določenih originalnih verzijah DOA uporabljajo nekatere interne parametre primerjalnih testov. Ker morajo instance problemov biti obravnavane, kot črne škatle (angl. black-boxes), so v platformi EDOLAB za vse algoritme DOA, ki uporabljajo parameter resnost premika (angl. shift severity), uporabili pri vseh isto metodo ocenitve resnosti premika in pri vseh, ki morajo za delovanje vedeti število obetavnih regij (angl. promising regions), so ta podatek nadomestili s številom pod-populacij. Rezultat primerjave so intervali zaupanja (angl. Rating Intervals - RI), ki jih dobimo za vsak DOA za vsako odigrano igro in jih lahko vidimo na sliki 1. Ker graf v naslednjih okoljih nadaljuje podoben trend, smo zaradi preglednosti prikazali le prvih 20 okolij problema. Na sliki je v obliki grafa prikazanih 101 interval zaupanja za vsak DOA, kjer ima višje postavljene DOA višji rating in je ocenjen kot uspešnejši od tistih, ki so na grafu pod njim. Glede na graf na sliki 1 se je algoritem mQSO tekom optimizacije MPB glede na druga dva algoritma odrezal najbolje. Vendar pri primerjanju dveh DOA ne moremo primerjati samo ratinga, ampak tudi interval zaupanja. Na sliki 2 smo prikazali graf za le prvih pet okolij, kjer je bolj razvidno, da se intervala mQSO in RPSO po 2000 ovrednotenjih več nikoli ne prekrivata, kar pomeni, da je med njima statistično značilna razlika v uspešnosti. Nasprotno, med dvema DOA ni statistično značilne razlike v uspešnosti, če se njuna intervala zaupanja prekrivata, kot lahko vidimo med 2000. in 7000. ovrednotenjem med algoritmoma DSPSO in RPSO. Bolj, kot se intervala prekrivata, manjša je razlika v uspešnosti, in ker gre za nedeterministične algoritme, bi se lahko tudi zgodilo, da bi se pri ponovnem zagonu eksperimenta vrstni red teh dveh

DOA spremenil. Na grafu so z navpično črtno črto označeni tudi trenutki, pri katerem ovrednotenju pride do spremembe okolja pri MPB. Na tak način lahko vidimo, da se uspešnost RPSO po vsaki spremembi nekoliko poslabša glede na preostala dva DOA.



Slika 2: Ratingi EDOA skozi prvih 5 okolij problema.

5 Zaključek

Ključna sposobnost DOA pri reševanju DOP je nenehno prilagajanje spremembam in tako učinkovito najti optimalno rešitev, kakor ji tudi tekom sprememb nenehno slediti. Ker se v literaturi pojavlja vedno več novih metod za reševanje DOP, je pomembno, da se predlagana nova metoda primerja z obstoječimi najsodobnejšimi algoritmi. V ta namen smo v prispevku predlagali novo metodo primerjave, ki temelji na šahovskem sistemu rangiranja Glicko-2. Rezultate smo z namenom lažje interpretacije konkurenčnosti algoritmov med celotnim postopkom optimizacije vizualizirali v obliki grafov. Prednost predlagane metode primerjave DOA je, da ni treba vnaprej poznati globalnega optimuma vsakega okolja, saj ta v večini dinamičnih problemov iz resničnega sveta ni znan vnaprej. Pomanjkljivost predlagane metode je ta, da so za določanje uspešnosti izbranega DOA v primerjavi potrebni drugi DOA.

Predlagano metodo je mogoče nadalje razširiti na algoritme dinamične optimizacije za reševanje večkriterijskih DOP tako, da namesto neposredne primerjave najdenih rešitev, primerjamo izračunane indikatorje kakovosti, ki se uporabljajo pri večkriterijski dinamični optimizaciji. Običajno se indikatorji kakovosti računajo na točki, preden pride do spremembe problema, vendar bi jih za našo primerjavo lahko računali tudi prej, na specifičnem ovrednotenju.

6 Zahvala

Raziskavo je finančno podprla Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije (ARIS) v okviru programov P2-0041 in P2-0114.

Literatura

[1] T. Blackwell and J. Branke. Multiswarms, exclusion, and anti-convergence in dynamic environments. *IEEE Tran-*

sactions on Evolutionary Computation, 10(4):459–472, 2006.

- [2] J. Branke. Memory enhanced evolutionary algorithms for changing optimization problems. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1875–1882 Vol. 3, 1999.
- [3] Jürgen Branke and Hartmut Schmeck. Designing Evolutionary Algorithms for Dynamic Optimization Problems. In Ashish Ghosh and Shigeyoshi Tsutsui, editors, *Advances in Evolutionary Computing: Theory and Applications*, pages 239–262. Springer, Berlin, Heidelberg, 2003.
- [4] Carlos Cruz, Juan R. González, and David A. Pelta. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing*, 15(7):1427–1448, July 2011.
- [5] Xiaohui Hu and R.C. Eberhart. Adaptive particle swarm optimization: detection and response to dynamic systems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 2, pages 1666–1670 vol.2, 2002.
- [6] Trung Thanh Nguyen, Shengxiang Yang, and Juergen Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24, October 2012.
- [7] Trung Thanh Nguyen and Xin Yao. Continuous Dynamic Constrained Optimization—The Challenges. *IEEE Transactions on Evolutionary Computation*, 16(6):769–786, December 2012. Conference Name: IEEE Transactions on Evolutionary Computation.
- [8] D. Parrott and Xiaodong Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation*, 10(4):440–458, 2006.
- [9] Mai Peng, Zeneng She, Delaram Yazdani, Danial Yazdani, Wenjian Luo, Changhe Li, Juergen Branke, Trung Thanh Nguyen, Amir H. Gandomi, Yaochu Jin, and Xin Yao. Evolutionary Dynamic Optimization Laboratory: A MATLAB Optimization Platform for Education and Experimentation in Dynamic Environments, August 2023. arXiv:2308.12644 [cs].
- [10] K. Trojanowski and Z. Michalewicz. Searching for optima in non-stationary environments. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1843–1850 Vol. 3, July 1999.
- [11] Niki Veček, Marjan Mernik, and Matej Črepinšek. A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms. *Information Sciences*, 277:656–679, September 2014.
- [12] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. A Survey of Evolutionary Continuous Dynamic Optimization Over Two Decades—Part A. *IEEE Transactions on Evolutionary Computation*, 25(4):609–629, August 2021. Conference Name: IEEE Transactions on Evolutionary Computation.
- [13] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. A Survey of Evolutionary Continuous Dynamic Optimization Over Two Decades—Part B. *IEEE Transactions on Evolutionary Computation*, 25(4):630–650, August 2021. Conference Name: IEEE Transactions on Evolutionary Computation.