

Izkoriščanje visokozmogljivega računalništva za porazdeljeno strojno učenje in optimizacijo hiperparametrov s pomočjo diferencialne evolucije

Teo Prica^{1,2}, Maja Pušnik², Aleš Zamuda²

¹Institut informacijskih znanosti, Prešernova ulica 17, 2000 Maribor, Slovenija

²Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija
E-pošta: teo.prica@izum.si, maja.pusnik@um.si, ales.zamuda@um.si

Harnessing High-Performance Computing for Distributed Machine Learning Hyperparameter Optimization Using Differential Evolution

From a scientific point of view, we are interested in how efficiently optimization techniques perform when we take into account their performance and consumption of computing resources, with an emphasis on energy-efficiency and sustainability in Artificial Intelligence (AI). Therefore, we have examined Differential Evolution (DE) optimization strategies in the context of machine learning (ML) of previously trained models and how they affect individual observation metrics when applied to real-world workflows within a high-performance computing (HPC) environment. Here, we want to highlight the importance of optimization and the efficiency of the selection of individual ML models and DE, taking into account the results obtained from the Slurm job scheduler. The experiment was conducted using the largest national petascale HPC.

1 Uvod

Novi pristopi, hitra rast in delovne obremenitve umetne inteligence, kjer kompleksnost narašča, zahteva precejšnjo količino računalniških virov, kar vodi v konvergenco od petaskalirne do po-eksaskalirne računalniške dobe [1]. Ker so podatkovni centri med največjimi porabniki in izpusti ogljika, se je pri tem ustvarila velika vrzel v prizadevanjih za prehod na bolj trajnostne pristope in strategije [2]. Prehod vodi do omejitev, kjer je potrebno uvesti učinkovite in temeljne mehanizme za trajnost [3, 4]. Zaradi različnih arhitektur in mehanizmov med sistemi se za zmanjšanje porabe računske moči lahko uporabijo učinkovitejši pristopi s ciljem zmanjšanja globalnega od-tisa [3]. Zelo pomembno vlogo imajo tudi uporabniki, ki

pridejo z različnim strokovnim znanjem iz različnih področij [5, 6]. Čeprav so na voljo tudi najboljše prakse in smernice, ki pomagajo pri učinkoviti in optimalni uporabi sistemov, uporabniki pogosto nimajo predhodnega znanja o HPC in morda ne dajejo prednosti učinkoviti uporabi in optimalnemu delovanju svojih delovnih obremenitev [7, 8]. Prav tako ni enotnega sistema za spremljanje in upravljanje porabe energije, saj se uporabljene tehnologije razlikujejo glede na posamezen sistem [7]. Dodatno so lahko metode, kot na primer ročno iskanje, naključno iskanje, Bayesova optimizacija, lahko časovno in računsko potratne, še posebej pri delu s kompleksnimi modeli in večjimi nabori podatkov [9–11].

V prispevku predstavljamo optimizacijo hiperparametrov konvolucijske nevronske mreže ResNet18 [12] na naboru podatkov MNIST in Fashion-MNIST, s pomočjo nadgradnje osnovnega populacijskega algoritma diferencialne evolucije (DE) [13]. Za nadzorovano strojno učenje in izvedbo zasnovanega eksperimenta smo uporabili delovni tok in predhodno naučene modele strojnega učenja, kjer smo izvedli dodatno uglaševanje modela na dodatnih podatkovnih zbirkah [14]. Dodatno smo uporabili delovni tok AutoDEHypO, ki je podrobneje predstavljen v članku [14]. S takšno nastavitvijo smo lahko uporabili uglaševanje različnih vidikov hiperparametrov. Tako predstavlja optimizacija hiperparametrov pomemben vidik strojnega učenja, kjer lahko iskanje optimalnega nabora parametrov izboljša delovanje delovnih obremenitev in modela [14]. Prvotno je bil delovni tok ovrednoten z uporabo predhodno naučenih modelov strojnega učenja, kot so ResNet18, DenseNet121, ConvNextSmall in VGG11, ob klasifikaciji slik z več oznakami na naboru podatkov zbirke CIFAR 10 in 100 [14, 15]. Delovni tok smo nadgradili in vključili dva dodatna nabora podatkov [16, 17]. Metoda nam omogoča izkoriščanje DE za učinkovito preiskovanje iskalnega prostora in ponuja rešitev za optimizacijo hiperparametrov modelov strojnega učenja [14]. Eksperiment smo izvedli z uporabo superračunalnika HPC Vega [5], kjer smo na izbrani infrastrukturi izkoristili vzporedno računanje na grafičnih procesorskih enotah (GPE), kar pospeši preiskovanje prostora hiperparametrov in skrajša skupni čas do rešitve [14].

Naslednji razdelki tega prispevka so organizirani na naslednji način. Razdelek 2 ponuja pregled sorodnih del in metod. Razdelek 3 opisuje metodologijo. Razdelek 4

Avtorji se zahvaljujejo EuroHPC JU, HPC RIVR in konzorciju SLING za dodelitev računalniških virov na nacionalnem deležu HPC Vega v okviru razvojnega projekta (S24R08-01), ki ga gosti Institut za informacijske znanosti (IZUM). Avtorji se zahvaljujejo projektu DAPHNE, ki ga financira program Evropske unije za raziskave in inovacije HORIZON 2020 v okviru sporazuma o dodelitvi nepovratnih sredstev št. 957407. Zahvaljujemo se tudi podpori programa COST (Evropsko sodelovanje v znanosti in tehnologiji) iz akcij COST: CA22137 "Raziskovalno omrežje naključnih optimizacijskih algoritmov (ROAR-NET)". Raziskava je bila izvedena v okviru samostojnega raziskovalnega dela znotraj učne enote Seminar 2 (55D006) doktorskega programa Računalništvo in informatika Univerze v Mariboru in delno financira IZUM (17-2375-2024/01-ab).

ponuja eksperimentalne rezultate. Razdelek 5 predstavlja zaključek in prihodnje delo.

2 Sorodna dela in metode

Optimizacija je zaporedje korakov, ki izboljšajo kakovost zadanega problema [18]. Probleme različnih razsežnosti in oblik najdemo v različnih domenah, kot je čas, dimenzija ali meritev, kjer so lahko problemi predstavljeni kot kriterijski vektorji glede na ovrednotenje rešitev v samem prostoru spremenljivk [18]. Dodatno je smiselno nasloviti kriterije, metode z utežmi in omejitve samih optimizacij, kot so metode z ohranjanjem dopustnih rešitev, uporaba kazenskih funkcij, s strogim razlikovanjem med dopustnostjo in nedopustnostjo vektorjev [18]. Optimizacijski algoritmi se uporabljajo v strojnem učenju za izboljšanje rezultatov, zlasti izgub v izbranem prostoru [18]. S pomočjo izbrane metode lahko optimiziramo hiperparametre še pred fazo učenja, kjer hiperparametri predstavljajo konfiguracijske spremenljivke ML modela [9, 10].

2.1 Superračunalniško okolje in programje

Superračunalnik je skupek večjega nabora med seboj povezanih vozlišč, ki zagotavljajo visoke zmogljivosti, katere merimo v številu operacij s plavajočo vejico na sekundo (FLOPS) in porabo energije v džulih (J) [5]. Uporabnost in izkoriščenost takšne infrastrukture narašča in predstavlja hrbtenico inovacij v slovenski in evropski akademski raziskovalni skupnosti kot tudi v industriji in manjših podjetjih [6]. HPC tako omogoča pohitritev in skrajšanje trajanja posameznih raziskav na različnih znanstvenih področjih vključno s strojnem učenjem [5, 6]. Za zagotavljanje potreb programske opreme v raziskovalni skupnosti in širše je znotraj uporabniškega okolja na voljo nabor optimiziranega in nameščenega programja, vključno s sistemskim programjem, vsebniki, knjižnicami in prevajalniki programske kode, ki so med drugim porazdeljeni in dostopni uporabnikom na slovenskih gručah preko oddaljenega datotečnega sistema CernVM-FS [5, 6]. Ob upoštevanju priporočil si manjkajoče različice programja lahko uporabniki pripravijo sami [6, 19]. Uporabnikom je prav tako na voljo večnivojska uporabniška podpora [5]. Za evropski delež je prav tako na voljo napredna aplikacijska podpora, ki je zagotovljena znotraj projekta EPI-CURE [20]. Naloge lahko uporabniki oddajo in upravljajo preko odprtokodne programske opreme Slurm [21].

Slurm je namenjen upravljanju in razporejanju z računalniškimi viri [21]. Programska oprema skrbi za okolje, pripravo nalog, čakalno vrsto, prioritete, ki vplivajo kdaj se bodo naloge izvedle v povezavi z linuxom (cgroups), poskrbi za omejitve pred, med in po izvajanju, skrbi za uporabniške račune ter njihovo porabo, nadzoruje izvajanje, vse dejavnosti posameznih izvajanj pa shrani v podatkovno bazo [21]. Omogoča prenosljivost glede na arhitekturo gruče, visok nivo odpornosti na napake, zagotavljanje varnosti vključno s hitro odpravo ranljivosti in pomankljivosti, dostopna je izvorna koda, ki omogoča hitro prilagodljivost in visok nivo skalabilnosti, preprostost glede na konfiguracijo in preprostost uporabe za končne uporabnike, kot tudi administratorje takšnih

sistemov, ki imajo na voljo dodatne funkcionalnosti [6, 21]. Pri tem pa Slovensko nacionalno superračunalniško omrežje (SLING) uporabnikom zagotavlja poenoteno programsko okolje znotraj gruč [5, 6]. Posamezni projekti imajo glede na tip projekta, kot na primer veliki, regularni, razvojni ali testni projekt, dodeljeno časovno omejitev trajanja projekta in kvoto računskih virov, ki jo lahko porabijo znotraj oddobrene alokacije [5]. Tako morajo uporabniki poskrbeti za učinkovito izrabo dodeljenih računskih moči, kjer bi morali prav tako posvetiti pozornost energijski učinkovitosti [8, 22]. Uporabniki tako oddajo svoje naloge na različne načine, bodisi preko ukaza *srn* ali predhodno pripravljene ter prenosljive skripte *bash* (SBATCH), ki zagotovi ponovljivost in avtomatizacijo ali interaktivno s pomočjo ukaza *salloc*, kjer se ustvari rezervacija na posameznem vozlišču vključno z interaktivno sejo [5, 6, 21].

2.2 Točka preverjanja in ponovni zagon

Točka preverjanja in ponovni zagon predstavljata mehanizem, za shranjevanje delovnih tokov [14]. Shranjene točke je mogoče pozneje znova zagnati, kar pomeni krajši čas izvajanja, manjšo porabo energije in možnost odpravljanja napak, ki se lahko pojavijo med izvajanjem delovne obremenitve [14]. Takšen mehanizem zagotavlja toleranco napak in odpornost na različne okvare, kot je na primer odpoved strojne opreme, napake na omrežju ali znotraj programske opreme [14], za dosego združljivosti raznolikega programja glede na izbrano arhitekturo, kot je na primer CPE in GPE, kot tudi vsebniška okolja, kjer je potrebna prilagoditev programske kode. [14].

2.3 Diferencialna evolucija

Algoritem diferencialne evolucije (DE) je del populacijskih algoritmov in sta ga leta 1995 predstavila Storn in Price [23]. DE se uporablja na področju optimizacije numeričnih funkcij in aplikacij znotraj različnih domenskih področij [23]. Osnovni DE ima glavno evolucijsko zanko, skozi katero se s pomočjo ključnih in osnovnih operatorjev, kot so inicializacija, mutacija, križanje in selekcija, po inicializaciji končni rezultat postopoma izboljšuje s ponavljanjem operatorjev, dokler se ne najde najboljša vrednost [23]. Na samem področju DE so tudi nenehne izboljšave, kot na primer Strukturirana diferencialna evolucija za zmanjšanje velikosti populacije z več strategijami mutacij (SPSRDEMMS), ki zagotavljajo statistično boljše rezultate kot podobni algoritmi, saj je zelo prilagodljiv [24]. Algoritem jDE uvaja razširitev osnovnega DE, kjer je dodan mehanizem, ki omogoča dinamično, boljše vrednosti prilagaja velikost populacijskega vektorja z iteracijami [25]. Adaptivno diferencialno evolucijo na podlagi zgodovine uspeha (SHADE) sta razvila Tanabe in Fukuaaga [26]. Algoritem SHADE se prilagaja glede na krmilne parametre, skaliranje (F) in križanje (CR) [26] in razširitve in optimizacija oziroma natančna nastavitve osnovnega algoritma so prispevale k uspešnim rezultatom v naslednjih letih, zato velja za učinkovit algoritem [26]. Dodan je bil mehanizem za linearno zmanjševanje populacije (LPSR) [27]. Obstajalo tudi izpeljani algoritmi, ki so v naslednjih letih prav tako zma-

gali na tekmovalstvu [26, 28]. Algoritem je prekosil druge algoritme tudi v resničnem okolju, kjer sta bili ocenjeni zanesljivost in učinkovitost uporabe DE v oceanskih misijah in načrtovanju podvodnih poti (UGPP) avtonomnih podvodnih vozil (AUV) [29]. Optimizacijski algoritem na razdalji temelječe prilagajanje krmilnih parametrov za diferencialno evolucijo, temelječo na uspehu (DISH) je razširitev prejšnjega algoritma L-SHADE, ki vključuje mehanizma za spreminjanje same velikosti populacije [18, 26]. Nadgradnja predhodno omenjenega algoritma DISH je še DISHv, kjer gre za podoben pristop kot prejšnji algoritem DISH, vendar imamo prenosljivo funkcijo z binarnimi parametri, kar posledično doprinese najboljši vektor za velikost danega problema (D) [18, 30].

2.4 Zelene izboljšave na področju strojnega učenja

Napredek in razvoj na področju avtomatskega strojnega učenja kažeta na potreben prispevek strategij in konceptov zelene umetne inteligence (UI) v povezavi z energijsko učinkovitostjo [2, 31]. Rdeča UI, ki je osredotočena na konvencionalne pristope ponovnega učenja modelov strojnega učenja in maksimiranje točnosti [2]. Kot rezultat pa naraste uporaba računskih virov, pri tem pa se zanemarljiva energijska učinkovitost [2]. Izboljšave in dokaz koncepta, ki temelji na knjižnici Scikit-learn in primerjava uporabljenih in potencialno primernih algoritmov s poudarkom na zeleni UI je predstavljena znotraj članka [2]. Geissler in sodelavci so predstavili nov ozaveščen pristop k optimizaciji hiperparametrov, ki temelji na zgodnji zavrnitvi neučinkovitih konfiguracij hiperparametrov za prihranek časa učenja strojnega učenja in porabe virov [32]. Evolucijski algoritmi so bili uporabljeni kot optimizatorji kot dodatek k AutoML, kot je izboljšan optimizacijski okvir z uporabo evolucijskih algoritmov [33]. Vakhnin in sodelavci so predstavili večciljni hibridni evolucijsko zasnovan okvir za natančno napovedovanje porabe energije [34]. Gomes in sodelavci so predstavili primerjalne študije, kjer je uporaba DE podajala boljše rezultate kot drugi algoritmi [35]. Prilagojenim študijam glede na specifična področja manjka integracija in uporaba HPC okolja (Slurm) [31]. Kljub zmogljivostim noben od podobnih orodij ne zagotavlja podpore za energijsko učinkovitost, saj se za merjenje porabe energije uporabijo dodatna orodja [14, 31].

2.5 Nabor podatkov

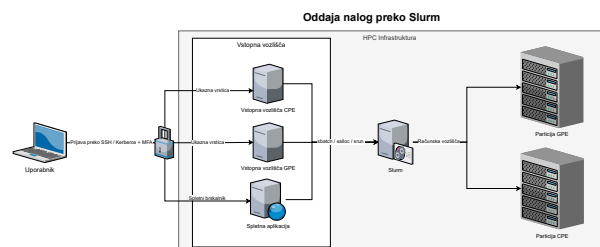
MNIST (Modified National Institute of Standards and Technology) je javna zbirka podatkov, ki vključuje ročne zapise števil [16]. Fashion-MNIST je javna zbirka zbranih sivinskih slik, ki prikazujejo različne kose oblačil, razdeljene v 10 razredov [17]. Zbirki MNIST in Fashion-MNIST vsebujeta vsaka po 60.000 slik (28 x 28) za strojno učenje in 10.000 slik za preizkušanje [16, 17].

3 Metodologija

Za nadzorovano strojno učenje in izvedbo zasnovanega eksperimenta smo uporabili predhodno naučene modele strojnega učenja, kjer smo izvedli dodatno ugaševanje modela na dodatnih podatkovnih zbirkah in nov primer

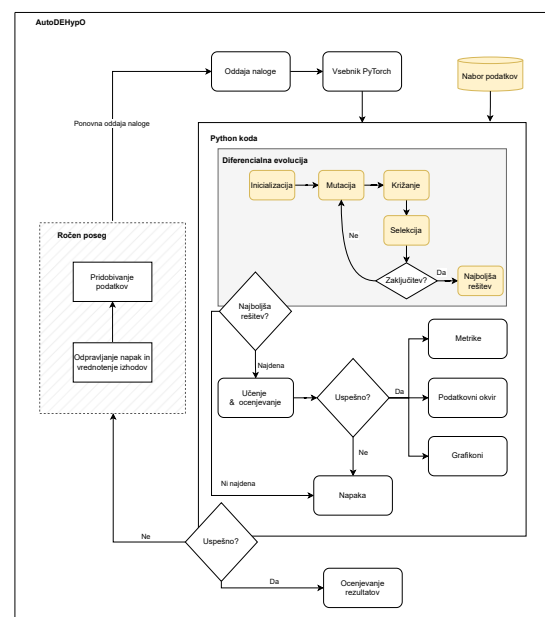
izkoriščanja delovnega toka AutoDEHypO. V tem novem primeru smo delovni tok nadgradili, uporabili kriterijsko funkcijo meritve napake za osnovno klasifikacijo in vključili dodatna nabora podatkov MNIST in Fashion-MNIST [16, 17]. Porabo računskih virov znotraj podatkovne zbirke beleži Slurm [21]. S pomočjo programa se beleži poraba časa, pomnilnika, virov, diskovnih kapacitet, porabljene energije, generičnih virov in številnih drugih. Slurm te podatke beleži za posamezne uporabniške račune, projekte in tudi posamezne naloge, kot je predstavljeno na sliki 1. Uporabniki lahko pridobijo podatke s pomočjo ukaza *sacct*. Primer takšnega ukaza, kjer *<jobid>* zamenjamo z dejanskim identifikatorjem naloge. Dodatno je možno prilagoditi parametre znotraj stikala *format*, kjer izberemo potrebne informacije.

```
sacct -u $USER -j <jobid> --format=\
JobID,ConsumedEnergy,CPUTime,elapsed,\
TresUsageOutAve,AllocTRES, \
partition --parsable2 --delimiter=";"
```



Slika 1: Osnovni delovni tok oddaje opravil preko Slurm.

Izbrali smo nedavno vzporedno implementacijo algoritma DE [36], ki vzporedno izvaja operacije s populacijo znotraj samega algoritma. Uporabljeno okolje prav tako vključuje vsebnik Singularity PyTorch, katerega delovni tok je predstavljen na sliki 2.



Slika 2: Shematski pregled delovnega toka AutoDEHypO.

Tabela 1: Rezultati po 10 epohah z uporabo modela ResNet18 na naboru podatkov MNIST (DE seed = 420).

Velikost serije	Najboljša točnost	Pretečen čas	Porabljena energija
16	0,2112	11 ur 42 min 00 s	19,30 MJ
32	0,2112	10 ur 28 min 00 s	17,31 MJ
64	0,2114	10 ur 20 min 38 s	17,41 MJ
128	0,2115	08 ur 53 min 48 s	15,69 MJ
256	0,2113	07 ur 10 min 03 s	11,88 MJ
512	0,2113	06 ur 43 min 59 s	11,58 MJ
1024	0,2112	08 ur 57 min 17 s	15,54 MJ

Tabela 2: Rezultati po 10 epohah z uporabo modela ResNet18 na naboru podatkov Fashion-MNIST (DE seed = 420).

Velikost serije	Najboljša točnost	Pretečen čas	Porabljena energija
16	0,1896	13 ur 40 min 20 s	22,79 MJ
32	0,1881	11 ur 11 min 20 s	18,55 MJ
64	0,192	10 ur 32 min 15 s	17,38 MJ
128	0,191	10 ur 47 min 36 s	18,30 MJ
256	0,1906	09 ur 30 min 43 s	16,20 MJ
512	0,1939	15 ur 03 min 21 s	24,87 MJ
1024	0,1922	11 ur 36 min 31 s	19,95 MJ

4 Rezultati

Za izvedbo poskusa smo uporabili superračunalnik HPC Vega [5]. Uporabljena so bila računska vozlišča na particiji z GPE. Celotno particijo sestavlja 60 vozlišč, kjer ima vsako vozlišče štiri NVIDIA Ampere 100 (A100) GPE, dva AMD Rome 7H12 procesorja, 512 GB pomnilnika, dve HDR mezzanine omrežni kartici z dvema priključkoma in 1,92TB M.2 SSD disk. Na vozliščih je nameščen operacijski sistem Red Hat Enterprise Linux 8.10, NVIDIA gonilniki verzije 565.57.01, CUDA verzije 12.7 ter razporejevalnik poslov Slurm 24.11.4 [5].

Rezultati enega zagona uglaševanja različnih vidikov hiperparametrov [37], kjer smo preliminarно opazovali število epoh in iteracij, uteži, hitrost učenja (LR), velikost serije in DE [36] kot optimizator, so predstavljeni znotraj tabel 1 in 2. Predstavljeni rezultati eksperimenta temeljijo na podlagi konvolucijske nevronske mreže ResNet in uporabljen ML model z 18 plastmi [12]. Glede na preliminarne opazovalne rezultate prilagoditve velikosti serije bistveno ne vplivajo na točnost izbranega ML modela, saj so točnosti precej podobne. Velikost serije lahko vpliva na izvajalni čas in porabo energije, kot na primer za nabor podatkov MNIST (tabela 1), kjer je optimalna velikost serije 512. Čas izvajanja je tako krajši tudi za približno 5 ur in porabi za 7,72 MJ manj energije (tabela 1). Vsekakor pa obstajajo izjeme, kjer velikost serije 512 na naboru podatkov Fashion-MNIST (tabela 2) lahko porabi več energije v primerjavi z drugimi zaključenimi nalogami, kot na primer za velikosti serij 64, 128, 256 in 1024. Tako zaznamo, da je za nabor podatkov Fashion-MNIST (tabela 2) optimalna velikost serije 256, kjer porabi do približno 1 uro manj izvajalnega časa pri tem pa porabi približno 1 MJ manj energije.

5 Zaključek in prihodnje delo

V prispevku je predstavljen nov primer izkoriščanja delovnega toka AutoDEHypO za optimizacijo hiperpara-

metrov, temelječ na diferencialni evoluciji. Za izvedbo zasnovanega eksperimenta smo uporabili predhodno naučene modele strojnega učenja, kjer smo izvedli dodatno uglaševanje modela pri uporabi zbirkah MNIST in Fashion-MNIST. Rezultati kažejo, da delovni tok v kontekstu HPC lahko zagotavlja ugoden kompromis in ravnoesje med natančnostjo in energetske učinkovitostjo. Dodatno omogoča prilagodljivost in učinkovit pristop k optimizaciji hiperparametrov, kjer ponuja prednosti kot tudi omejitve izrabe računskih virov, zaznavanje strojnih okvar v HPC okolju in granularnost ob merjenju zadanih metrik. Spremljanje in poročanje o porabljeni energiji, ter izkoriščanje razpoložljivih mehanizmov za zmanjšanje porabe energije se je izkazalo za izvedljivo.

V prihodnje bo delo usmerjeno na izboljšanje prilagodljivosti algoritma širšemu naboru modelov strojnega učenja, dodatnih naborov podatkov, preizkus drugih arhitektur ter sistemov in vključevanje dodatnih naprednih tehnik, kot je večkriterijska optimizacija. Izkoriščanje tovrstnih infrastruktur bo v prihodnje ključnega pomena. Pri tem bi tudi izpostavili omejitve, ki so vezane na dodeljene računske vire znotraj posameznega projekta.

Literatura

- [1] T. Prica in A. Zamuda, "Monitoring Energy Consumption of Workloads on HPC Vega," V *Proceedings of the 6th ISC HPC International Workshop on Monitoring & Operational Data Analytics*, Hamburg, Germany, 2025.
- [2] V. Bolón-Canedo, L. Morán-Fernández, B. Cancela in A. Alonso-Betanzos, "A review of green artificial intelligence: Towards a more sustainable future," *Neurocomputing*, zv. 599, stran 128096, 2024.
- [3] C. Silva, R. Vilaça, A. Pereira in R. Bessa, "A review on the decarbonization of high-performance computing centers," *Renewable and Sustainable Energy*, zv. 189, 2024.
- [4] A. Alonso-Betanzos, *Rethinking Efficiency in Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2025.
- [5] T. Prica, "Development in supporting activities on Euro-HPC Vega," V *Austrian-Slovenian HPC Meeting 2024 — ASHPC24*, 2024.
- [6] D. Lesjak, A. Prah in B. Krašovec, "From Isolation to Integration: A Decade of Container Technology in Slovenian HPC," V *Nordic e-Infrastructure Tomorrow*, A. Azab in T. Malkiewicz, Ured. Cham: Springer Nature Switzerland, 2025, str. 142–151.
- [7] R. Iakymchuk, G. Gedik, K. Kulkarni, Y. Chen, D. Kempf, S. Kemmler, D. Papageorgiou, D. Konioris, S. Kiebdaj, J. Corbalan in H. Köstler, "Best Practice Guide – Harvesting energy consumption on European HPC systems: Sharing Experience from the CEEC project," 2024.
- [8] E. Suarez, J. Amaya, M. Frank, O. Freyermuth, M. Girone in B. Kostrzewa, "Energy efficiency trends in HPC: what high-energy and astrophysicists need to know," *Frontiers in Physics*, zv. 13, 2025.
- [9] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A.-L. Boulesteix, D. Deng in M. Lindauer, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining and Knowledge Discovery*, zv. 13, št. 2, stran 1484, 2023.

- [10] A. Morales-Hernández, I. Van Nieuwenhuysen in S. Rojas Gonzalez, "A survey on multi-objective hyperparameter optimization algorithms for machine learning," *Artificial Intelligence Review*, zv. 56, stran 8043–8093, 2022.
- [11] J. Bergstra, R. Bardenet, Y. Bengio in B. Kégl, "Algorithms for hyper-parameter optimization," V *Proceedings of the 25th International Conference on Neural Information Processing Systems*, ser. NIPS'11. Curran Associates Inc., 2011, stran 2546–2554.
- [12] K. He, X. Zhang, S. Ren in J. Sun, "Deep Residual Learning for Image Recognition," 2015.
- [13] R. Storn in K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, zv. 11, št. 4, stran 341–359, 1997.
- [14] T. Prica in A. Zamuda, "High-Performance Deployment Operational Data Analytics of Pre-trained Multi-label Classification Architectures with Differential-Evolution-Based Hyperparameter Optimization (AutoDEHypO)," *Mathematics*, zv. 13, št. 10, stran 1681, 2025.
- [15] "Dataset CIFAR10 in CIFAR100," <https://www.cs.toronto.edu/~kriz/cifar.html>, Dostopno 2025/07/18.
- [16] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, zv. 29, št. 6, str. 141–142, 2012.
- [17] H. Xiao, K. Rasul in R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," 2017.
- [18] A. Zamuda, *Operacijske raziskave logističnih, transportnih in ekonomskih sistemov: zbrano gradivo*. Fakulteta za elektrotehniko, računalništvo in informatiko, 2020.
- [19] B. Krasovec in T. Prica, "Secure Usage of Containers in the HPC Environment," V *Nordic e-Infrastructure Tomorrow*, A. Azab in T. Malkiewicz, Ured. Cham: Springer Nature Switzerland, 2025, str. 96–112.
- [20] A. Prah, "EPICURE and containers, activities in EuroHPC," *ASHPC25: Austrian-Slovenian HPC Meeting 2025*, stran 47, 2025.
- [21] A. B. Yoo, M. A. Jette in M. Grondona, "Slurm: Simple linux utility for resource management," V *Workshop on job scheduling strategies for parallel processing*. Springer, 2003, str. 44–60.
- [22] A. Chinnici, E. Ahmadzada, A.-L. Kor, D. De Chiara, A. Domínguez-Díaz, L. de Marcos Ortega in M. Chinnici, "Towards sustainability and energy efficiency using data analytics for hpc data center," *Electronics*, zv. 13, št. 17, 2024.
- [23] S. Das, S. S. Mullick in P. N. Suganthan, "Recent Advances in Differential Evolution – An Updated Survey," *Swarm and Evolutionary Computation*, zv. 27, 2016.
- [24] A. Zamuda, J. Brest in E. Mezura-Montes, "Structured Population Size Reduction Differential Evolution with Multiple Mutation Strategies on CEC 2013 real parameter optimization," V *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 2013, str. 1925–1931.
- [25] J. Brest, S. Greiner, B. Boskovic, M. Mernik in V. Zumer, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, zv. 10, št. 6, str. 646–657, 2006.
- [26] A. V. in Roman Senkerik in Michal Pluhacek in Tomas Kadavy in Ales Zamuda, "Distance based parameter adaptation for Success-History based Differential Evolution," *Swarm and Evolutionary Computation*, zv. 50, stran 100462, 2019.
- [27] M. F. Ahmad, N. A. M. Isa, W. H. Lim in K. M. Ang, "Differential evolution: A recent review based on state-of-the-art works," *Alexandria Engineering Journal*, zv. 61, št. 5, str. 3831–3872, 2022.
- [28] P. N. Suganthan, "P-N-Suganthan GitHub Public repository," <https://github.com/P-N-Suganthan/>, Dostopno 2025/05/18.
- [29] A. Zamuda in J. D. H. Sosa, "Success history applied to expert system for underwater glider path planning using differential evolution," *Expert Systems with Applications*, zv. 119, str. 155–170, 2019.
- [30] A. Zamuda, "Binary 100-digit challenge using IEEE-754 coded numerical optimization scenarios (100b-digit) and V-shape binary distance-based success history differential evolution (DISHv)," V *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2019, str. 1821–1828.
- [31] M. Baratchi, C. Wang, S. Limmer, J. N. van Rijn, H. Hoos, T. Bäck in M. Olhofer, "Automated machine learning: past, present and future," *Artificial Intelligence Review*, zv. 57, št. 5, 2024.
- [32] D. Geissler, B. Zhou, S. Suh in P. Lukowicz, "Spend More to Save More (SM2): An Energy-Aware Implementation of Successive Halving for Sustainable Hyperparameter Optimization," 2024.
- [33] A. M. Vincent in P. Jidesh, "An improved hyperparameter optimization framework for AutoML systems using evolutionary algorithms," *Scientific Reports*, zv. 13, št. 1, 2023.
- [34] A. Vakhnin, I. Ryzhikov, H. Niska in M. Kolehmainen, "A Novel Multi-Objective Hybrid Evolutionary-Based Approach for Tuning Machine Learning Models in Short-Term Power Consumption Forecasting," *AI*, zv. 5, št. 4, stran 2461–2496, 2024.
- [35] E. Gomes, L. Pereira, A. Esteves in H. Morais, "Metaheuristic Optimization Methods in Energy Community Scheduling: A Benchmark Study," *Energies*, zv. 17, št. 12, stran 2968, jun 2024.
- [36] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt in SciPy 1.0 Contributors, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, zv. 17, št. 3, str. 261–272, 2020.
- [37] D. Vermetten, H. Wang, M. López-Ibañez, C. Doerr in T. Bäck, "Analyzing the impact of undersampling on the benchmarking and configuration of evolutionary algorithms," V *Proceedings of the Genetic and Evolutionary Computation Conference*, 2022, str. 867–875.