Very Large Scale Global Optimization with Randomised Optimisation Algorithms in DAPHNE

Aleš Zamuda

University of Maribor
Faculty of Electrical Engineering and Computer Science
Institute of Computer Science
Koroška cesta 46, 2000 Maribor, Slovenia
ales.zamuda@um.si

Globalna optimizacija zelo velikih dimenzij z naključenimi optimizacijskimi algoritmi v DAPHNE

Povzetek — Ta članek predstavlja globalno optimizacijo zelo velikih dimenzij (GOZVD) z naključenimi optimizacijskimi algoritmi (NOA) v DAPHNE (Integracija kanalov za analizo podatkov za upravljanje velikih podatkov, HPC in strojno učenje). Najnovejša programska oprema iz DAPHNE repozitorija `daphne` na računu GitHub `daphne-eu`, z uporabo zadnje veje `main` iz julija 2024 (548ea01), je prevedena in nameščena na superračunalniku EuroHPC Vega v Mariboru, Sloveniji. Namestitev je nato pognana z uporabo Slurma za izvedbo niza zagonov ROA z različnimi razredi nastavitev, ob povečevanju števila optimiziranih parametrov na sto tisoč. Nazadnje so poročani in analizirani še novo dobljeni računski rezultati iz zagonov ROA.

Abstract — This paper presents the Very Large Scale Global Optimization (VLSGO) in context of Randomised Optimization Algorithms (ROA) in DAPHNE (Integrated Data Analysis Pipelines for Large-Scale Data Management, HPC, and Machine Learning). The latest software from DAPHNE repository `daphne` at the GitHub account `daphne-eu`, using the last July 2024 commit to the `main` branch (548ea01), is compiled and deployed on EuroHPC Vega supercomputer in Maribor, Slovenia. The compilation is then deployed using Slurm to execute a set of ROA runs with different configuration classes for VLSGO, increasing the optimized parameter sizes. Finally, the newly obtained computational results from ROA runs are reported and analysed.

1 Introduction

This paper presents the Very Large Scale Global Optimization (VLSGO) in context of Randomised Optimization Algorithms (ROA) in DAPHNE (Integrated Data Analysis Pipelines for Large-Scale Data Management, HPC, and Machine Learning) [1–4]. VLSGO as characterized in this paper is with ten thousand floating-point encoded parameters dimension sizes and above, where ROA with large population sizes are applied as well (here, in size of a thousand). When benchmarking ROAs, they also need to be run multiple times, with independent and different random number seeds, because ROAs can be statistically evaluated only with aggregated statistics from these independent runs [5], to further use this in Machine Learning [6].

Some results were already presented at [7, 8] and are now extended to VLSGO and more recent DAPHNE software. Progress on High-Performance Computing (HPC), with DAPHNE, and Computational Intelligence (CI, as defined through IEEE CIS [9]) in Slovenia has also previously been reported in e.g. [10–12], and with EuroHPC Vega the use of HPC with latest CI and DAPHNE has become more accessible. Therefore, this paper benchmarks the latest progress in ROA and DAPHNE with Vega, where latest July 2024 compilation, deployment, and results from DAPHNE is performed, where a ROA algorithm is executed on VLSGO.

In the next section, the related work is described, then the methods are described in Section 3. Section 4 provides the results and Section 5 the conclusion.

2 Related Work

ROA in DAPHNE has already been identified as an opportunity to scale structures like matrices in [10] and an implementation was also deployed on EuroHPC Vega [7], implemented in DaphneDSL [13]. Additionally to be a partner in the DAPHNE project, the decision to choose a HPC DSL is substantiated further especially after considering that University of Maribor is part of SLING (Slovenian National Supercomputing Network), the coordination body of various HPCs in Slovenia, including the first EuroHPC supercomputer [14], the Vega with 240 A100 GPUs and 122 thousand CPU (Central Processing Unit) cores [15]. More about how to deploy ROA in DAPHNE on EuroHPC was already presented in [7] and more about DAPHNE in Maribor is in [4, 7, 10, 12, 15–17].

The DaphneDSL syntax [13] is inspired by C/Java-like languages and the DSL is a case-sensitive language inspired by ML systems as well as languages and libraries

This work is supported by project DAPHNE (Integrated Data Analysis Pipelines for Large-Scale Data Management, HPC, and Machine Learning) funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 957407, locally coordinated by AZ at University of Maribor (UM, www.um.si). AZ also gratefully acknowledges the HPC RIVR consortium (www.hpc-rivr.si) and EuroHPC JU (eurohpc-ju.europa.eu) for also funding this research by providing computing resources (S24V01-03) of the HPC system Vega at the Institute of Information Science (www.izum.si).

for numerical computation like Julia, Python NumPy, R, and Apache SystemDS DML [18], but also with compiler hints for data/operator placement (i. e. local/distributed, CPU/GPU/FPGA, computational storage, still in experimental state and not guaranteed by the compiler) [13].

2.1 Population-based Randomised Optimization Algorithms

Population-based ROAs are surveyed in works like [19–26], with several CI applications like [24,27–30]. A sample ROA consists of e.g. an evolutionary loop [31], within which are evolved new population D-dimensional population vectors $\mathbf{x}_i$, $\forall i \in \{1, 2, ..., NP\}$. During each generation step number $g \in \{1, 2, ..., G\}$, on the population, computational operators are performed like mutation, crossover, and selection, until a termination criterion is satisfied, like a fixed number of maximum fitness evaluations (*MAX_FES*). ROA has also been winning competitions at GECCO [32] and applied in context of HPC in works like [33], where a large number of fitness evaluations was executed. Also, parallelization of benchmarking using HPC, from text summarization in natural language processing, glider piloting in deep-sea missions, and search algorithms in computational intelligence has been presented in [12] and then reflected for generative AI in [11]. As indicated further for [34], “Traditionally, single objective benchmark problems are also the first test for new evolutionary and swarm algorithms,” therefore in this paper a challenging single-objective optimisation function HappyCat [35] is applied to test a ROA on VLSGO, where the fitness evaluation of a numerical input vector $\mathbf{x}$ is computed by the function $f = \left(\left(\sum\left(x^2\right) - 10\right)^2\right)^{0.125} + \left(\sum\left(x^2\right)/2 + \sum\left(x\right)\right)/10 + 0.5$.

3 Methods: VLSGO with ROA in DAPHNE

The latest software from DAPHNE repository `daphne` at the GitHub account `daphne-eu` using the last July 2024 commit to the `main` branch `548ea01` as seen in Figure 2, is compiled as seen in Figure 3 taking several hours to compile and deployed on EuroHPC Vega supercomputer in Maribor, Slovenia. This compilation is also possible much faster due to CMake, parallel in minutes, if adding more compute resources, while recompilations during development of reprogrammed code parts are down to seconds and well supported in open or closed source code editors like Vim and Qt Creator, or Visual Studio Code. The DaphneDSL code snippet to evaluate a fitness function f is provided in Figure 1: the arithmetics is listed in line 2, while the function definition and input type parameter x is defined as `matrix<f64>`, a 64-bit floating-point matrix, processed with the DaphneDSL kernel operations (see [7] for their list in f on CPU).

The compilation is deployed using Slurm and large maximum stack size, to execute a set of ROA runs with different configuration classes for VLSGO (line 1: scaled from scale level $S = 1$ to $S = 10$), increasing the optimized parameter sizes (line 3: dimensions per ten thousand) and the allowed runtime according to this increase (line 8: 288 minutes per scaling level S, which is 28880 minutes for $D = 100.000$, i.e. two days for the largest configurations expecting to run the longest) as seen in Figure 4. If rerun with same parameters, algorithm results are reproducible and provide same output numbers.

4 Results

The optimisation results from the ROA runs (fitness convergence through generations) as explained in the above deployment preparation, are presented in Figure 5 as a set of convergence graphs, in configurations with dimensions $D = 1000$ and population sizes NP from 10.000 to 100.000 in increases of 10.000 as the scaled parameter classes (S from 1 to 10), capped up to the allocated time limit for each run.

Considering that the allocated time limit might stop an execution of a task, the plots in Figure 5 ran with this timing stopping condition up to their allocated time, all jobs started, and produced useful output results. The plots are drawn after all tasks have executed the allocated runs. As the runs with dimension ten thousand were running the longest, to cap the runtimes on the `cpu` partition, the multiplier of 288 minutes per scale S was used as shown in Figure 4 on line 8. All jobs have successfully reached target generations of 300 for the VLSGO tasks, except 4 cancelled earlier by Slurm due to timeout on node cn0514 (jobs 31746064 on $S = 5$, $RNi = 6$ at $G = 298$; 31746065 on $S = 3$, $RNi = 6$ and $RNi = 1$ both at $G = 289$) and cn0570 (31746113, $G = 269$).

For each of the runs plotted, it is observed that the fitness function optimised by the ROA is successfully improving, hence, the ROA CI is performing its main functionality of optimisation.

Further, using Universal Machine Learning Analysis Utility (Umlaut) from GitHub `https://github.com/daphne-eu/umlaut`, the resource usage is tracked during execution. As an example, for configuration $D = 10.000$, $NP = 1000$, the total runtime over these independent runs is seen in Figure 6. Furthermore, Figures 7 and 10 provide usage of memory and CPU resources, respectively. The memory usage plot peeks at approximately 105.87 MB for these runs, by initially rising to roughly 89 MB and then staying almost all of the time at that usage, slightly varying because of iterative allocations. As memory was seen increasing during the run, in meanwhile after experiments have already been done in August using lates July code updates and by the time of revision of this paper in September, also for loops [17], the implementation of allocations has been upgraded in the language implementation with newest

```
1 def eval_f(x: matrix<f64>) -> matrix<f64> {
2     return ( (sum(x * x, 0) - 10) ^ 2 ) ^ 0.125 + ( sum(x * x, 0) / 2 + sum(x) ) / 10 + 0.5;
3 }
```

Figure 1: Fitness function f evaluation snippet in DaphneDSL.

```
git clone https://github.com/daphne-eu/daphne.git
```

Figure 2: Cloning the DAPHNE main repository from daphne-eu repository of daphne-eu at GitHub, from July 26, 2024.

```
[DAPHNE]..Successfully built Daphne://daphne (took 4h 37min 624ms 452us 620ns)........................[31.07.24 15:22:43]
```

Figure 3: Output feedback after compiling the latest codes on EuroHPC Vega using a single Slurm task.

```
for S in {1..10}; do
 for NP in 1000; do
  D=${S}0000
  echo D=$D NP=$NP
  for RNi in {1..10}; do
   echo -n .
   { time srun --mpi=none \
    --time $((288*$S)) --mem=${S}G \
    ../daphneeu.sif \
    ./run-daphne.sh roa.d \
    D=$D NP=$NP RNi=$RNi \
    > results-D-$D-NP-$NP-RNi-$RNi-out.txt &
   } 2> time-D-$D-NP-$NP-RNi-$RNi-out.txt
  done #RNi
 done #NP
done #D

wait
```

Figure 4: Deploying ROA on EuroHPC Vega with Slurm, scaling the optimized dimension sizes.

software release (version 0.3) and additional regular updates.

A plotting of run times over scaling level S for D is seen in Figures 8 and 9, where runtime dash-dotted linear fit of data points is seen increasing with S. For $RNi = 1$ (the first run) of $D = 10.000$ (smallest S) and $RNi = 4$ (typical, median run) of $D = 100.000$ (largest S), plots for CPU load are drawn in Figure 10. CPU load is baselined at 100% (one thread) and the jitters with higer loads are attributed to multi-threaded executions of DAPHNE kernels with Basic Linear Algebra Subprograms (BLAS). Deviations in running time among independent runs were observed and could be studied further.

5 Conclusion

This paper presented a pipeline benchmarking with latest July 2024 deployment of DAPHNE (Integrated Data Analysis Pipelines for Large-Scale Data Management, HPC, and Machine Learning) on EuroHPC Vega, running ROA CI tasks with Slurm on VLSGO (Very Large Scale Global Optimization). An example ROA benchmarking pipeline scenario using HappyCat function was benchmarked and VLSGO scaling results were discussed, together with resource usage as obtained through Umlaut. All runs have successfully been running, optimizing fitness, and their provided resource usage reported.

Future work still includes further deployment (e.g. to additional hardware and improved scheduling of workload in distributed multi-node Slurm tasks), comparisons of benchmarking results [11, 12], integrated benchmarking, benchmarking using updated versions of DAPHNE software, and extending the ROA scenario and library as well as research in other use cases, especially from CI and remote sensing, including underwater missions like ocean glider path planning and text summarization.

References

[1] Publication Office of the European Union, "Fact sheet : Integrated data analysis pipelines for large-scale data management, hpc, and machine learning," in *CORDIS – EU research results*, 2024, p. https://cordis.europa.eu/project/id/957407.

[2] P. Damme, M. Birkenbach, C. Bitsakos, M. Boehm, P. Bonnet, F. Ciorba, M. Dokter, P. Dowgiallo, A. Eleliemy, C. Faerber, G. Goumas, D. Habich, N. Hedam, M. Hofer, W. Huang, K. Innerebner, V. Karakostas, R. Kern, T. Kosar, A. Krause, D. Krems, A. Laber, W. Lehner, E. Mier, M. Paradies, B. Peischl, G. Poerwawinata, S. Psomadakis, T. Rabl, P. Ratuszniak, P. Silva, N. Skuppin, A. Starzacher, B. Steinwender, I. Tolovski, P. Tözün, W. Ulatowski, Y. Wang, I. Wrosz, A. Zamuda, C. Zhang, and X. X. Zhu, "Daphne: An open and extensible system infrastructure for integrated data analysis pipelines," in *12th Conference on Innovative Data Systems Research, CIDR 2022, Chaminade, CA, January 9-12*, 2022.

[3] A. Vontzalidis, S. Psomadakis, C. Bitsakos, M. Dokter, K. Innerebner, P. Damme, M. Boehm, F. Ciorba, A. Eleliemy, V. Karakostas, A. Zamuda, and D. Tsoumakos, "DAPHNE Runtime: Harnessing Parallelism for Integrated Data Analysis Pipelines," in *Euro-Par 2023: Parallel Processing Workshops*, ser. Lecture Notes in Computer Science, vol. 14352, D. Zeinalipour, D. B. Heras, G. Pallis, H. Herodotou, D. Trihinas, D. Balouek, P. Diehl, T. Cojean, K. Fürlinger, M. H. Kirkeby, M. Nardellli, and P. D. Sanzo, Eds. Cham: Springer, 2024, pp. 242–246.

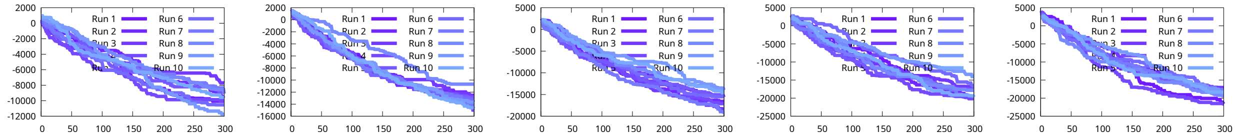
[4] T. Tomažič and A. Zamuda, "Digital Strategic Communication: the Case of the 1st DAPHNE Symposium," in *Proceedings of 33rd International Electrotechnical and Computer Science Conference*, 2024, p. (in print).

[5] A. Zamuda, *Operacijske raziskave logističnih, transportnih in ekonomskih sistemov: zbrano gradivo*. Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, 2020.

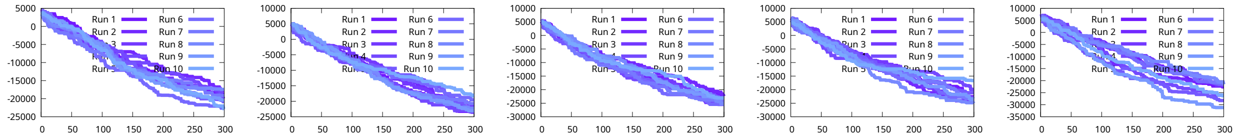
[6] ——, "Business intelligence through computational intelligence for data science: an overview of the breadth of real numerical optimization solutions – through autonomous submarines, forest modeling, power plant management, space probes, protein modeling, radio signal coding to text summarization," in *lecture presented at SMART FUTURE Data Science and Business Intelligence, UNI.MINDS, 17. november 2020*, 2020.

[7] A. Zamuda and M. Dokter, "Deploying DAPHNE Computational Intelligence on EuroHPC Vega for Benchmarking Randomised Optimisation Algorithms," in *International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom)*, 2024, p. 8.

[8] B. Steinwender, V. Jancauskas, A. Laber, M. Birkenbach, B. Einberger, D. Krems, and A. Zamuda, "Integrated Data Analysis Pipelines for Large-Scale Data Management, HPC, and Machine Learning : D8.3 Benchmarking Results all Use Case Studies,"



(a) $D = 10.000, NP = 1000$ (b) $D = 20.000, NP = 1000$ (c) $D = 30.000, NP = 1000$ (d) $D = 40.000, NP = 1000$ (e) $D = 50.000, NP = 1000$



(f) $D = 60.000, NP = 1000$ (g) $D = 70.000, NP = 1000$ (h) $D = 80.000, NP = 1000$ (i) $D = 90.000, NP = 1000$ (j) $D = 100.000, NP = 1000$

Figure 5: Convergent optimisation runs (fitness on vertical axis, number of executed generations on horizontal axis) using DAPHNE ROA on function HappyCat, for different independent seeds, with D from $D = 10.000$ (case (a)) upto $D = 100.000$ in case (j).

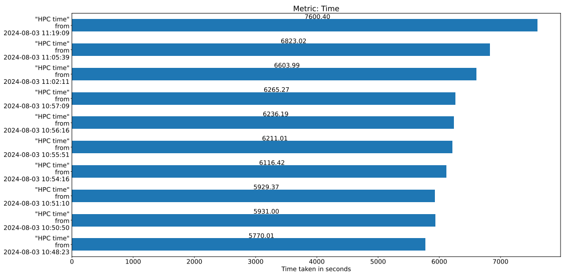


Figure 6: Time duration for DAPHNE ROA on function HappyCat, $D = 10.000$, $NP = 1000$, over 10 RNi configurations, measured through an Umlaut pipeline.

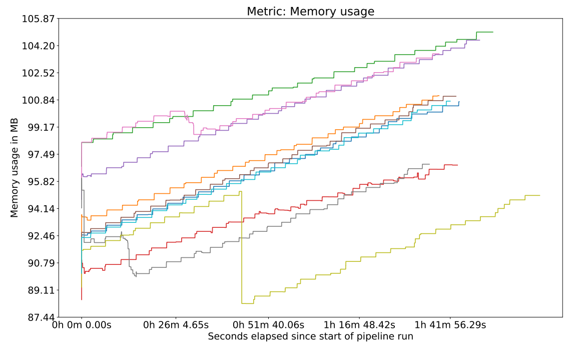


Figure 7: Memory usage profiles during the run execution of DAPHNE ROA on function HappyCat, $D = 10.000$, $NP = 1000$, and 10 RNi configuration values.

KAI (KAI Kompetenzzentrum Automobil- und Industrieelektronik GmbH), DLR (Deutsches Zentrum für Luft- und Raumfahrt EV), IFAT (Infineon Technologies Austria AG), AVL (AVL List GmbH), and UM (Univerza v Mariboru), Tech. Rep. Version 2.1, 2024.

[9] G. B. Fogel, "A History of the IEEE Computational Intelligence Society," in *2009 IEEE Conference on the History of Technical Societies*. IEEE, 2009, pp. 1–5.

[10] A. Zamuda, "Randomised Optimisation Algorithms in DAPHNE," in *Austrian-Slovenian HPC Meeting 2024 – ASHPC24*, 2024, p. 33.

[11] ——, "Generative AI Using HPC in Text Summarization and Energy Plants," in *Austrian-Slovenian HPC Meeting 2023– ASHPC23*, 2023, p. 5.

[12] ——, "Parallelization of benchmarking using HPC: text summarization in natural language processing (NLP), glider piloting in deep-sea missions, and search algorithms in computational intelligence (CI)," in *Austrian-Slovenian HPC Meeting 2021 - ASHPC21*, 2021, p. 35.

[13] DAPHNE: An Open and Extensible System Infrastructure for Integrated Data Analysis Pipelines, "DaphneDSL Language Reference," in *GitHub repository daphne-eu/daphne*, 26 July 2024. [Online]. Available: https://github.com/daphne-eu/daphne/blob/main/doc/DaphneDSL/LanguageRef.md

[14] A. Zamuda, "Monitoring and operational data analytics from a user perspective at first EuroCC HPC Vega supercomputer and nation-wide in Slovenia," in *invited lecture at moda21 : Second International Workshop on Monitoring and Operational Data Analytics, 2 July 2021, Strasbourg (France)*, 2021.

[15] MIZŠ, "Ministrstvo za izobraževanje, znanost in šport (Direktorat za znanost in inovacije): HPC Vega kot temelj za inovacijsko-raziskovalni projekt DAPHNE," in *Portal GOV.SI : spletišče državne uprave s celovitimi informacijami o njenem delovanju in preprostim dostopom do storitev. Ljubljana: Urad Vlade Republike Slovenije za komuniciranje. 5. 1. 2022*, 2022, pp. https://www.gov.si/novice/2022–01–05–hpc–vega–kot–temelj–za–inovacijsko–raziskovalni–projekt–daphne/.

[16] M. Divjak and A. Zamuda, "Experimental pipeline definition for surface high-density electromyogram (HDEMG) processing," in *Proceedings of 33rd International Electrotechnical and Computer Science Conference*, 2024, p. (in print).

[17] B. Bošković, J. Brest, and A. Zamuda, "Loops of the Domain-specific Programming Language DaphneDSL," in *Proceedings of 33rd International Electrotechnical and Computer Science Conference*, 2024, p. (in print).

[18] M. Boehm, I. Antonov, S. Baunsgaard, M. Dokter, R. Ginthör, K. Innerebner, F. Klezin, S. Lindstaedt, A. Phani, B. Rath, B. Reinwald, S. Siddiqi, and S. B. Wrede, "SystemDS: A Declarative Machine Learning System for the End-to-End Data Science Lifecycle," in *10th Annual Conference on Innovative Data Systems Research, CIDR 2020*, 2020.

[19] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[20] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[21] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82–117, 2013.

[22] A. Zamuda and J. Brest, "Self-adaptive control parameters' randomization frequency and propagations in differential evolution," *Swarm and Evolutionary Computation*, vol. 25, pp. 72–99, 2015.

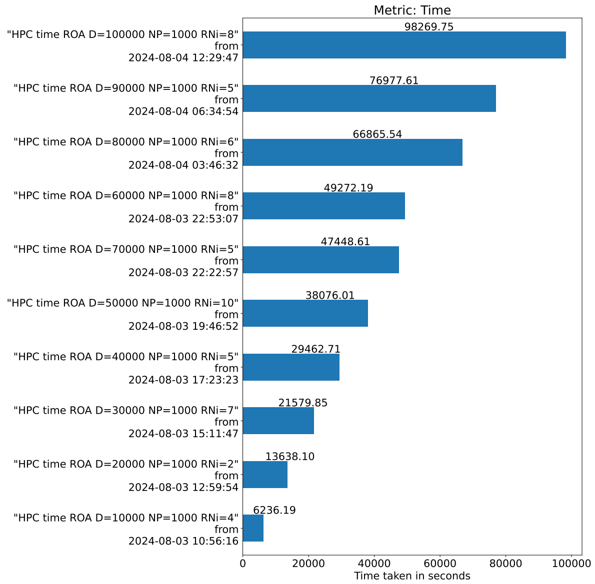


Figure 8: Runtime reported for DAPHNE ROA on function HappyCat, $NP = 1000$, for some selected runs with different D. As D increases, the runtimes are also increasing.

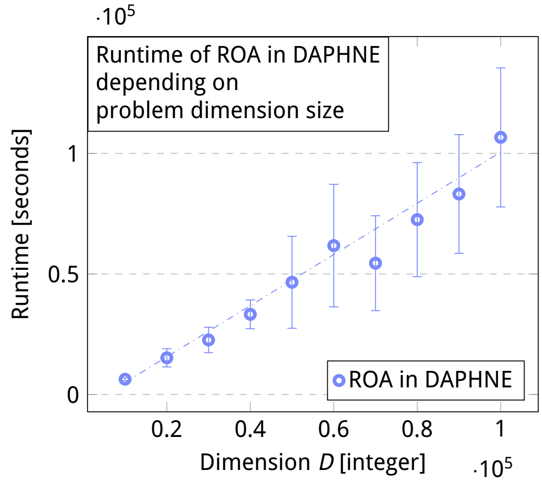
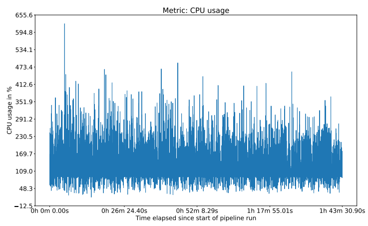
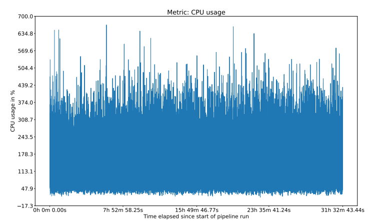


Figure 9: Runtime reported for DAPHNE ROA on function HappyCat, depending on problem dimension size (D), $NP = 1000$, drawn as average and standard deviations.



(a) $D = 10.000, RNi = 1$



(b) $D = 100.000, RNi = 5$

Figure 10: CPU load during run execution using DAPHNE ROA on function HappyCat $NP = 1000$, for case a) $D = 10.000$, $RNi = 1$ and $D = 100.000$, $RNi = 5$.

[23] K. Sörensen, "Metaheuristics—the metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015.

[24] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution – An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[25] A. P. Piotrowski, "Review of differential evolution population size," *Swarm and Evolutionary Computation*, vol. 32, pp. 1–24, 2017.

[26] X. Liu, J. Sun, Q. Zhang, Z. Wang, and Z. Xu, "Learning to learn evolutionary algorithm: A learnable differential evolution," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, pp. 1605–1620, 2023.

[27] A. Zamuda and J. Brest, "Vectorized procedural models for animated trees reconstruction using differential evolution," *Information Sciences*, vol. 278, pp. 1–21, 2014.

[28] A. Glotić and A. Zamuda, "Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution," *Applied Energy*, vol. 141, pp. 42–56, 1 March 2015.

[29] R. P. Parouha and K. N. Das, "Dpd: An intelligent parallel hybrid algorithm for economic load dispatch problems with various practical constraints," *Expert Systems with Applications*, vol. 63, pp. 295–309, 2016.

[30] A. Zamuda and J. D. H. Sosa, "Success history applied to expert system for underwater glider path planning using differential evolution," *Expert Systems with Applications*, vol. 119, no. 1 April 2019, pp. 155–170, 2019.

[31] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[32] A. Zamuda, "Function Evaluations Upto 1e+12 and Large Population Sizes Assessed in Distance-based Success History Differential Evolution for 100-Digit Challenge and Numerical Optimization Scenarios (DISH-chain1e+12): A competition entry for "100-Digit Challenge, and Four Other Numerical Optimization Competitions" at The Genetic and Evolutionary Computation Conference (GECCO) 2019," in *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic*, 2019, pp. ACM, New York, NY, USA.

[33] ——, "Solving 100-digit challenge with score 100 by extended running time and parallel benchmarking," in *SOR '23 : proceedings of the 17th International Symposium on Operational Research in Slovenia : Bled, Slovenia, September 20-22, 2023. 1st electronic version. Ljubljana: Slovenian Society Informatika - Section for Operational Research*, 2023, pp. 206–211.

[34] K. V. Price, N. H. Awad, M. Z. Ali, and P. N. Suganthan, "The 100-Digit Challenge: Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization," Vacaville, California, USA and School of EEE, Nanyang Technological University, Singapore and School of Computer Information Systems, Jordan University of Science and Technology, Jordan, Technical Report, Nanyang Technological University, Singapore, November 2018, 2018.

[35] H.-G. Beyer and S. Finck, "HappyCat – A Simple Function Class Where Well-Known Direct Search Algorithms Do Fail," in *International conference on parallel problem solving from nature*. Springer, 2012, pp. 367–376.