SustainML

Application Aware, Life-Cycle Oriented Model-Hardware Co-Design Framework for Sustainable, Energy Efficient ML Systems

Carbon footprint based model optimization tool

Deliverable D3.1

WP3 - Energy Consumption Optimized ML Toolkit and Methods

This project has received funding from the European Union's Horizon Europe research and innovation programme (HORIZON-CL4-2021-HUMAN-01) under grant agreement No 101070408

eProsima
The Middleware Experts

DFKI

Inria

upmem

RPTU Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

IBM

Project

Title:	SustainML: Application Aware, Life-Cycle Oriented Model-Hardware Co-Design Framework for Sustainable, Energy Efficient ML Systems
Acronym:	SustainML
Coordinator:	eProsima
Grant agreement ID:	101070408
Call:	HORIZON-CL4-2021-HUMAN-01
Program:	Horizon Europe
Start:	01 October 2022
Duration:	36 months
Website:	https://sustainml.eu
E-mail:	sustainml@eprosima.com
Consortium:	**eProsima (EPROS)**, Spain **DFKI**, Germany **Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU)**, Germany **University of Copenhagen (KU)**, Denmark **National Institute for Research in Digital Science and Technology (INRIA)**, France **IBM Research GmbH**, Switzerland **UPMEM**, France

Deliverable

Number:	**D3.1**
Title:	**Carbon footprint based model optimization tool**
Month:	6
Work Package:	WP3 - Energy Consumption Optimized ML Toolkit and Methods
Work Package leader:	KU
Deliverable leader:	KU
Deliverable type:	R, DEM
Dissemination level:	Public (PU)
Date of submission:	2023-03-31
Version:	v1.0
Status:	Finished

Version history

Version	Date	Responsible	Author/Reviewer	Comments
v1.0	31-03-2323	KU	KU	

Executive summary

Energy consumption from selecting, training and deploying deep learning models has grown exponentially over the past few years. Our goal in this work is to support the design of energy-efficient deep learning models that are easier to train with low compute resources, practical to deploy in real-world edge/mobile computing settings and environmentally sustainable. While tabular benchmarks allow the evaluation of Neural Architecture Search (NAS) strategies at low computational cost by providing pre-computed performance statistics, today's NAS algorithms and benchmarks focus mainly on standard performance measures like accuracy. We suggest including energy efficiency as an additional performance criterion to NAS and extending an existing tabular benchmark by including information on energy consumption and carbon footprint. The benchmark called EC-NAS is open-source to support energy consumption-aware NAS research. We demonstrate the usefulness of EC-NAS by applying multi-objective optimisation algorithms that uncover the trade-off between energy consumption and accuracy, showing that it is possible to discover energy-efficient architectures with high accuracies.

This report corresponds to *Deliverable D3.1 - Carbon footprint based model optimization tool* of the SustainML project. This deliverable covers an overview of the growing energy consumption problems in deep learning techniques and the importance of energy-efficient models for real-world applications. A summary of existing NAS methods, tabular benchmarks, and the current focus on performance as the primary objective for model development, and why energy efficiency and carbon footprint are crucial performance/efficiency criteria to consider.

Additionally, this report outlines the approach for extending a tabular benchmark to incorporate energy consumption and carbon footprint metrics during the training and evaluation of deep learning models, introducing the EC-NAS benchmark. The application of multi-objective optimisation algorithms to EC-NAS and the utilisation of the metrics above for energy-efficient model development are discussed. Results and analysis are presented, emphasising the trade-offs between energy consumption and performance and potential use cases for real-world applications. Finally, future work and areas for improvement are suggested. For full details on EC-NAS and elaborate discussions on the topics above, see [1].

Contents

1 Introduction

NAS strategies explore predefined search spaces for potential model architectures, determining fitness based on validation/test set performance to discover the best-performing architecture [2]. Despite success in finding novel designs [3, 4, 5, 6], NAS's high computational cost, energy consumption, and significant carbon footprint are drawbacks [7, 8, 9, 10]. Tabular benchmarks and surrogate models have improved NAS evaluation efficiency [11, 12, 13, 14, 15, 5, 16], but the focus remains on performance over efficiency.

In this delivable, we incorporate energy consumption as an additional criterion in tabular NAS benchmarks to discover energy-efficient models for practical deployment and sustainability, presenting EC-NAS [1]. Optimizing energy consumption and performance show good promise in revealing computationally efficient models with minimal performance loss and smaller carbon footprints.

2 Energy Consumption Awareness

NAS for efficient architectures has primarily focused on optimising run-time or floating point operations (FPOs) [17]. However, FPOs may not fully represent model efficiency [18, 19, 20]. Recently, energy consumption-optimised hyperparameter selection was studied outside NAS settings for large language models [21]. Energy consumption during model training encompasses aspects not covered by standard resource constraints like FPOs, computational time, and the number of parameters. It accounts for hardware and software variations, making it a suitable additional objective for NAS.

Roughly 75% of total energy costs when training neural networks come from hardware accelerators like GPUs, and TPUs [9, 22]. The rest is mainly due to CPUs and DRAM, with supporting infrastructure accounted for by power usage effectiveness (PUE). Open-source tools like experiment-impact-tracker [18], Carbontracker [9], and CodeCarbon [23] help track energy consumption. In EC-NAS, the energy consumption is estimated with Carbontracker [9], monitoring GPUs, CPUs, and DRAM to determine total energy costs, E (kWh), carbon footprint ($kgCO_2eq$), and total computation time, T(s) summarised in Table 1.

Metrics	Unit of measurement	Notation
Model parameters	Million (M)	$\|\theta\|$
Test/Train/Eval. time	Seconds (s)	$T(s)$
Test/Train/Val. Acc.	$\mathbb{R} \in [0; 1]$	P_v
Energy consumption	Kilowatt-hour (kWh)	E(kWh)
Power consumption	Joule (J), Watt (W)	E(J), E(W)
Carbon footprint	$kgCO_2eq$	–
Carbon intensity	g/kWh	–

Table 1: Metrics reported in `EC-NAS-Bench`.

3 EC-NAS

In EC-NAS, 423k unique architectures are examined with training budgets of 4, 12, 36, and 108 epochs to explore potential trade-offs between performance and resource expenditure. Following the NAS-Bench-101 approach, each model is trained on CIFAR-10 using 40k training samples and validated and tested on 10k samples (60k in total). Every model is trained using an in-house Slurm cluster with a single

NVIDIA Quadro RTX 6000 GPU, 24 GB memory, and two Intel CPUs. The training strategy, including hyperparameter settings, aligns with NAS-Bench-101 [11].

Predicting energy consumption from a few training epochs is reliable on the same hardware [9]. Instead of re-training all NAS-Bench-101 models, we select a subset of 4300 models (approximately 1% of the total space) and obtain their actual energy costs for 4 epochs of training and evaluation, the minimum queryable performance statistic in the original dataset. This allows for augmenting the dataset with efficiency measure predictions for full training budgets while maintaining accurate performance estimates from the original dataset. Using these measurements, we train a multi-layered perceptron (MLP) based surrogate energy prediction model, which is discussed further next.

3.1 Surrogate Energy Predictor

The MLP-based surrogate energy model takes the graph-encoded architecture and the number of parameters as input, predicting energy consumption for a given number of epochs. This surrogate model is similar to current surrogate NAS methods that have efficiently facilitated NAS evaluations on larger search spaces and provided more accurate performance estimates than tabular benchmarks [14]. Surrogate models are used as one-time compute methods for cost-effective evaluation of NAS methods within extensive search spaces and do not apply to other search spaces.

The MLP-based surrogate model used for predicting the training energy consumption of the 7V space, E, is given as: $f_\theta(\cdot) : \mathbf{x} \in \mathbb{R}^F \rightarrow E \in \mathbb{R}$, where θ are the trainable parameters and $\mathbf{x}$ consists of the F features obtained from architecture specifications. We populate $\mathbf{x}$ with the upper triangular entries of the adjacency matrix, operations mapped to categorical variables and the total number of parameters. For the 7V space, this results in $\mathbf{x} \in \mathbb{R}^{36}$.

The surrogate energy model is implemented as a simple four-layered MLP with gelu activation functions and trained using actual energy measurements from 4310 randomly sampled architectures in the 7V space. The model is implemented in Pytorch and trained on a single NVIDIA RTX 3060 GPU. The training, validation, and test split has a ratio of [0.7,0.1,0.2], resulting in [3020,430,860] data points, respectively. Using the Adam optimiser, the MLP is trained for 200 epochs with an initial learning rate of 5×10^{-3} to minimise the L1-norm loss function between the predicted and actual energy measurements.

The resulting surrogate dataset closely approximates the actual training energy costs, with a Pearson correlation of 0.9977 between actual and predicted energy measurements on the test set (left) Figure 1. A high correlation is expected, considering the search space exhibits a high degree of locality, especially for smaller models. The mean absolute error of predicted and actual energy measurements plateaus when trained with about 3000 architectures (right), justifying its use for predicting the remaining space.

4 NAS strategies with EC-NAS

With a tabular benchmark that allows querying for model training energy consumption alongside other standard metrics, as EC-NAS, NAS strategies can be employed to discover energy-efficient architectures. We now introduce multi-objective optimisation as an appropriate strategy for revealing the trade-off between performance and efficiency, facilitating an informed, energy-conscious architecture selection.

4.1 Multi-Objective Optimisation

Multi-objective optimisation (MOO) concurrently optimises multiple, possibly conflicting objectives. MOO aims to identify or approximate the set of Pareto-optimal solutions. A solution is deemed Pareto-optimal if it cannot be improved in one objective without deteriorating in another. The usefulness of

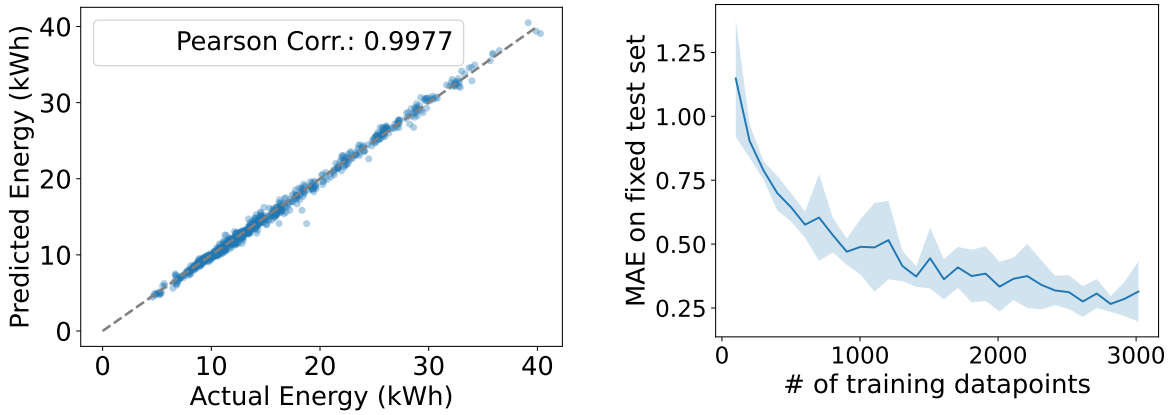

Figure 1: Scatter plot (left) showing the correlation between the predicted and actual energy consumption for the test set obtained from a subset of the architectures and (right) influence of the number of training data points on the test set performance in the Surrogate model. Error bars are computed over 10 random initialisations.

EC-NAS is demonstrated when performing MOO using several algorithms, including a simple evolutionary MOO algorithm (SEMOA) based on [24]. The algorithm is simple but derived from canonical principles of derivative-free multi-criteria optimisation, such as hypervolume maximisation. We also use other existing MOO algorithms: Speeding up Evolutionary Multi-Objective Algorithms (SHEMOA) and Mixed Surrogate Expected Hypervolume Improvement (MS-EHVI) implemented in [25] to demonstrate the usefulness of EC-NAS and compare results to baseline Random Search (RS) procedure.

4.2 Experimental Setup

We conduct experiments on EC-NAS by performing both single-objective optimisation (SOO) and MOO. In the former, we will naturally find only one solution when optimising a single objective, which is done using RS. Experiments are conducted on the entire model space generated using the surrogate energy predictor, considering models for all epoch budgets $e \in 4, 12, 36, 108$. Optimisation runs for a maximum of 100 generations for the evolutionary algorithms with a population size of 10, and for RS, a total of 1000 function evaluations. All experiments are carried out for 10 trials on a single workstation equipped with an 11th generation Intel Core i7-1185G7 @ 3.00GHz $\times$ 8 processor and integrated TGL GT2 graphics, running Ubuntu 22.04.1 LTS.

4.3 Performance Criteria

For the multi-objective optimisation, we use the validation accuracy (P_v) and the training energy cost, E(kWh), as the two objectives to be jointly optimised using the MOO algorithm. We use energy cost rather than, e.g., training time, considering that E is agnostic to parallel computing. We note that using any of the provided metrics in Table 1 for both single- and multi-objective optimisations is possible. As the MOO algorithm minimises the objectives, we use the negative of the objectives in cases where the quantities are to be maximised; for instance, we optimise $1 - P_v$ as accuracy is a maximisation objective.

Model	Strategy	$\lvert\theta\rvert$(M)↓	$T(s)\downarrow$	$P_v\uparrow$	E(kWh)↓
$\mathcal{A}_{\mathbf{r}_0}$ (B)	SEMOA	7.26 ±0.15	2555.95 ±202.42	0.94 ±0.01	0.62 ±0.08
$\mathcal{A}_{\mathbf{r}_1}$ (G)	SEMOA	**5.95** ±0.05	**14.23**	0.52 ±0.52	**0.0** ±0.01
$\mathcal{A}_{\mathbf{r}_k}$ (Y)	SEMOA	6.43 ±0.06	306.9 ±41.86	0.92 ±0.01	0.07 ±0.01
$\mathcal{A}_{\mathbf{r}^*}$ (R)	SOO	7.05 ±0.18	1649.11 ±342.02	**0.94** ±0.01	0.41 ±0.09

Table 2: Metrics for models in single- and multi-objective setting seen in Figure 2. For SOO the optimal solution ($\mathcal{A}_{\mathbf{r}^*}$/Red) is reported. For MOO the two extrema ($\mathcal{A}_{\mathbf{r}_0}$/Blue, $\mathcal{A}_{\mathbf{r}_1}$/Green) and the knee point ($\mathcal{A}_{\mathbf{r}_k}$/yellow) are reported.

5 Results

The key results from the experiments on EC-NAS using the multi-objective optimisation of E and $1-P_v$ are shown in Figure 2. The Pareto fronts over multiple random initialisations of the four MOO algorithms: SEMOA (ours), Random Search, SHEMOA, MSEHVI, are visualised as attainment curves (left) which summarises the median solutions attained over the multiple runs [26]. All the MOO algorithms can explore the search space reasonably well, yielding similar attainment curves.

The Pareto front obtained from our MOO algorithm, SEMOA, for one run is shown in Figure 2 (right). It also shows the extrema ($\mathbf{r}_0, \mathbf{r}_1$) on both ends of the front preferring one of the objectives, whereas the knee point ($\mathbf{r}_k$) offers the best trade-off between the two objectives. These three points are shown in different colours and markers, where the two extrema ($\mathcal{A}_{\mathbf{r}_0}$/Blue, $\mathcal{A}_{\mathbf{r}_1}$/Green) and the knee point ($\mathcal{A}_{\mathbf{r}_k}$/yellow).

The architectures corresponding to the two extrema ($\mathcal{A}_{\mathbf{r}_0}$,$\mathcal{A}_{\mathbf{r}_1}$) and the corresponding knee point ($\mathcal{A}_{\mathbf{r}_k}$) overall MOO runs are visualised in the radar plot in Figure 2 (middle). The exact performance metrics for thereof are also reported in Table 2. The solution covering the most significant area is one of the extremal points ($\mathcal{A}_{\mathbf{r}_0}$, blue) with high accuracy (0.94) but also a larger footprint in the energy consumption (0.41kWh), computation time (2555.95s) and the number of parameters (7.26M) compared to the other extremum ($\mathcal{A}_{\mathbf{r}_1}$, green) or the knee point ($\mathcal{A}_{\mathbf{r}_k}$, yellow). The model corresponding to the knee point ($\mathcal{A}_{\mathbf{r}_k}$) provides a significant reduction in the energy consumption (0.07kWh) at the expense of a slight reduction in performance (0.92).

6 Conclusion

In conclusion, this work has presented EC-NAS, a novel neural architecture search (NAS) benchmark that incorporates energy consumption awareness, expanding upon the traditional focus on performance metrics. Combining this additional objective with multi-objective optimisation (MOO) strategies allows energy-efficient architectures to be identified while maintaining competitive performance. Adopting an energy-aware approach in NAS is crucial for mitigating the environmental impact of training large-scale neural networks, as hardware accelerators contribute significantly to energy consumption. EC-NAS enables researchers to explore the trade-offs between performance, energy consumption, and other standard metrics, supporting informed decision-making in architecture selection and ultimately promoting the development of more sustainable AI systems.

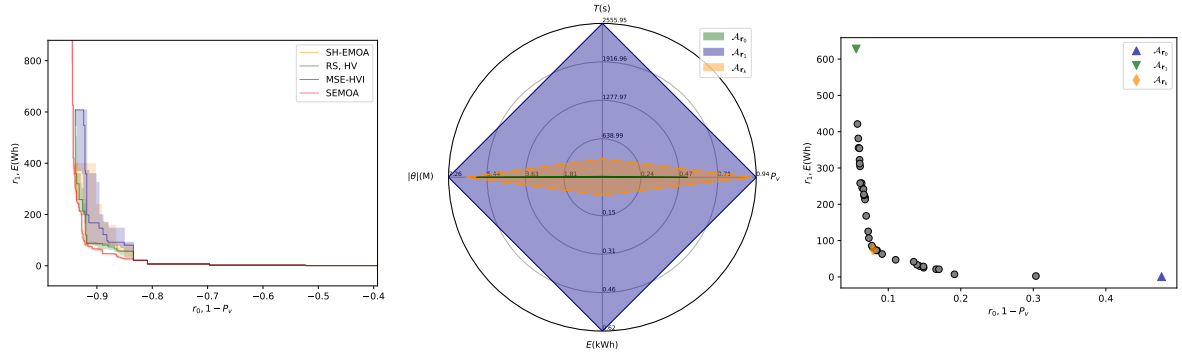

Figure 2: Multi-objective exploration: (left) shows the attainment curve showing the median solutions and the inter-quartile variations for 10 random initialisations of the four MOO algorithms on the EC-NAS benchmark. The two objectives being optimised $(1 - P_v, E)$ are shown on the two axes. Average architecture metrics (middle) shown correspond to the two extrema ($\mathcal{A}_{\mathbf{r}_0}$ in blue, $\mathcal{A}_{\mathbf{r}_1}$ in green) and the knee point ($\mathcal{A}_{\mathbf{r}_k}$ in yellow). The plot has four axes capturing the performance P_v and resource consumption measured in $E, T, |\theta|$. Additionally, the Pareto front (right) obtained for one of the MOO runs shows the family of solutions as discrete points.

References

[1] Pedram Bakhtiarifard, Christian Igel, and Raghavendra Selvan. "Energy Consumption-Aware Tabular Benchmarks for Neural Architecture Search". In: *arXiv preprint arXiv:2210.06015* (2022).

[2] Pengzhen Ren et al. *A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions*. 2020. DOI: 10.48550/ARXIV.2006.02903. URL: https://arxiv.org/abs/2006.02903.

[3] Chenxi Liu et al. *Progressive Neural Architecture Search*. 2017. DOI: 10.48550/ARXIV.1712.00559. URL: https://arxiv.org/abs/1712.00559.

[4] Hanxiao Liu, Karen Simonyan, and Yiming Yang. *DARTS: Differentiable Architecture Search*. 2018. DOI: 10.48550/ARXIV.1806.09055. URL: https://arxiv.org/abs/1806.09055.

[5] Ming Lin et al. "Zen-NAS: A Zero-Shot NAS for High-Performance Deep Image Recognition". In: *International Conference on Computer Vision (ICCV)*. 2021.

[6] Bowen Baker et al. *Accelerating Neural Architecture Search using Performance Prediction*. 2017. DOI: 10.48550/ARXIV.1705.10823. URL: https://arxiv.org/abs/1705.10823.

[7] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: (2019). DOI: 10.48550/ARXIV.1905.11946. URL: https://arxiv.org/abs/1905.11946.

[8] Roy Schwartz et al. *Green AI*. 2019. DOI: 10.48550/ARXIV.1907.10597. URL: https://arxiv.org/abs/1907.10597.

[9] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. *Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models*. ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems. arXiv:2007.03051. July 2020.

[10] Jaime Sevilla et al. "Compute trends across three eras of machine learning". In: *arXiv preprint arXiv:2202.05924* (2022).

[11] Aaron Klein and Frank Hutter. *Tabular Benchmarks for Joint Architecture and Hyperparameter Optimization*. 2019. DOI: 10.48550/ARXIV.1905.04970. URL: https://arxiv.org/abs/1905.04970.

[12] Xuanyi Dong and Yi Yang. "NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search". In: *International Conference on Learning Representations (ICLR)*. 2019.

[13] Wei Wen et al. *Neural Predictor for Neural Architecture Search*. 2019. DOI: 10.48550/ARXIV.1912.00848. URL: https://arxiv.org/abs/1912.00848.

[14] Arber Zela et al. "Surrogate NAS Benchmarks: Going Beyond the Limited Search Spaces of Tabular NAS Benchmarks". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=OnpFa95RVqs.

[15] Chris Ying et al. "NAS-Bench-101: Towards Reproducible Neural Architecture Search". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, Sept. 2019, pp. 7105–7114. URL: http://proceedings.mlr.press/v97/ying19a.html.

[16] Arber Zela, Julien Siems, and Frank Hutter. *NAS-Bench-1Shot1: Benchmarking and Dissecting One-shot Neural Architecture Search*. 2020. DOI: 10.48550/ARXIV.2001.10422. URL: https://arxiv.org/abs/2001.10422.

[17] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[18] Peter Henderson et al. "Towards the systematic reporting of the energy and carbon footprints of machine learning". In: *Journal of Machine Learning Research* 21.248 (2020), pp. 1–43.

[19] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[20] Yunho Jeon and Junmo Kim. "Constructing fast network through deconstruction of convolution". In: *Advances in Neural Information Processing Systems* 31 (2018).

[21] Lucas Høyberg Puvis de Chavannes et al. "Hyperparameter Power Impact in Transformer Language Model Training". In: *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*. Virtual: Association for Computational Linguistics, Nov. 2021, pp. 96–118. DOI: 10.18653/v1/2021.sustainlp-1.12. URL: https://aclanthology.org/2021.sustainlp-1.12.

[22] Jesse Dodge et al. "Measuring the carbon intensity of ai in cloud instances". In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022, pp. 1877–1894.

[23] Victor Schmidt et al. "CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing". In: (2021). DOI: 10.5281/zenodo.4658424.

[24] Oswin Krause, Tobias Glasmachers, and Christian Igel. "Multi-objective Optimization with Unbounded Solution Sets". In: *NeurIPS Workshop on Bayesian Optimization (BayesOpt 2016)*. 2016.

[25] Sergio Izquierdo et al. "Bag of baselines for multi-objective joint neural architecture search and hyperparameter optimization". In: *8th ICML Workshop on Automated Machine Learning (AutoML)*. 2021.

[26] Viviane Grunert da Fonseca, Carlos M Fonseca, and Andreia O Hall. "Inferential performance assessment of stochastic optimisers and the attainment function". In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2001, pp. 213–225.