Representing Experience in Continuous Evolutionary Optimisation through Problem-tailored Search Operators

Stephen Friess[§], Peter Tiňo[§], Stefan Menzel[†], Bernhard Sendhoff[†] and Xin Yao[§‡]
[§] CERCIA, School of Computer Science, University of Birmingham, UK
[†] Honda Research Institute Europe GmbH, 63073 Offenbach a.M., Germany
[‡] Southern University of Science and Technology, Shenzhen, China
{shf814, p.tino, x.yao}@cs.bham.ac.uk, {stefan.menzel, bernhard.sendhoff}@honda-ri.de

***Abstract*—Evolutionary algorithms are a class of population-based metaheuristic methods partially inspired by natural evolution. Specifically, they rely on stochastic variation and selection processes to sequentially find optimal solutions of a function of interest. We attempt in this work to extract preferences in these stochastic evolutionary operators in form of empirical and improved distributions as basis for model-based mutation operators. The latter can be considered as representing problem-tailored search operators which exist independently from the optimisation run and thus can be transferred to similar problem instances. This offline approach is different to existing model-based optimisation techniques, e.g. EDA's, CMA-ES and Bayesian approaches, where adaption happens rather in an online manner without the influence of prior experience. Our approach can be rather considered to follow the recent line of research on knowledge transfer in optimisation, which until now heavily relies upon the transfer of candidate solutions across different optimisation tasks. We investigate in this paper the interplay between algorithm and optimisation task, its influence on the retrieved distributions and explore whether or not these can lead to performance improvements on a selected range of problems, as well as when transferring them across problems. At last, we make a comparison of built distributions in the hope of relating similarity in statistical distances between distributions to possible performance gains.**

***Index Terms*—Evolutionary computation, metaheuristic optimization, statistical machine learning, knowledge transfer.**

I. Introduction

Evolutionary algorithms are notable for being metaheuristics, meaning they try to avoid being problem specific. Nevertheless, most of them come with a set of various hyperparameters and degrees of freedoms which have to be chosen by the practitioner first. In this case, the correct choice of parameters can indeed have a beneficial and problem-dependent effect on the performance of the algorithm. An interesting, however barely explored topic to this regard is whether or not the efficiency of an algorithm can be directly improved from prior optimisation experiences by means of harnessing statistical properties arising in the algorithm-problem interaction. In our work, we try to address the issue of learning from prior solved problem instances by means of improving the search operators for new problem instances. While in principle, one may question whether this approach defeats the purpose of a metaheuristic in the first place, one can counter this argument with the fact that generalizing this method could allow one to design specialized operators for entire classes of problems directly from previous experience. Introducing learning in a metaheuristic context therefore does not defeat its purpose, but instead allows one to integrate it into a framework of higher level heuristics [1].

The remainder of this paper is structured as follows. We first review in Sec. II relevant literature in the domain of metaheuristic stochastic optimisation and give an overview on methods used which try to establish a notion of learning or knowledge transfer. Secondly, in Sec. III we introduce the algorithm of interest for our study and explain necessary modifications. These will enable us to build a distribution for improved sampling in our framework. We note at this point, our approach is different from notions of learning found in adaptive optimisation algorithm such as EDAs [2], CMA-ES [3] and Bayesian approaches (e.g. [4]), as well as recent advances on transfer learning in optimisation [5]–[9] which operate on the sole basis of direct or indirect solution transfer. This is because we explicitly try to transfer knowledge of optimisation runs from a procedural view by means of building a static distribution for sampling, which represents rough and globally averaged properties of a fitness landscape. It is also different in the sense of local improvements or Lamarckian learning, because resampled mutations from this distribution may not necessarily be improving for individual candidate solutions. In our conducted experiments, we investigate first the algorithm-problem interaction by looking at the obtained search distribution while varying free parameters in the algorithm configuration and problem of interest, as well as investigate the dynamics and how the sampling width influences the performance of the retrieved search distribution. At last, we try to relate algorithm performance with different statistical distances between the obtained distributions. We conclude this paper in Sec. IV with a summary of our study, highlight peculiarities of our framework and give an outlook for potential future work.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

II. Related Work

First notable steps towards a learning framework for metaheuristics were made in 2004 through the CIGAR framework [10]. Specifically, it proposes a case-based approach where intermediate and final solutions from previously solved combinatorial optimisation tasks are kept in a storage. Whenever a new optimisation problem is tackled, this case-base is queried and solutions are sampled probabilistically from similar previously tackled problems under the further help of a task-similarity measure based upon problem characteristics. These then sub-sequentially form part of the initial population on the new task of interest. A more recent approach similar in the sense of the case-base from CIGAR is also represented by AMTEA [5]. Within their work, Gaussian distributions are used to model final populations from previously tackled continuous multi-objective optimisation tasks. When new tasks are encountered, periodically a mixture model is constructed from the repository to approximate the current generation in the new optimisation task of interest. The obtained weights of the mixture model are used subsequently to sample probabilistically new child solutions from the previously solved tasks.

Note that their work reflects a problem similarity through solution similarity philosophy. This has been originally proposed by [10] as a way to cope with situations where task similarity measures are not trivially definable. Further notable works also concern the repeating construction of a linear mapping between ranked intermediate solutions of a task, which is then subsequently used to map final or current best solution of a past or concurrent related task into the population [6]–[8]. Note, that the former assume that for effectiveness of their method, task similarity and thus complementarity can be or has been established a priori. A similar approach has been also used in the dynamic multi-objective optimisation algorithm Tr-DMOEA ([9], [11]). However, aside from a few works [12], there exists no clear concept on what constitutes task similarity in a more abstract manner. And secondly, how to learn task specific characteristics in a generalized way and from a procedural perspective, to help improve solving future similar tasks without relying on an explicit solution transfer.

In reflection upon these unanswered questions in the literature, we will introduce in the following a simple framework for continuous evolutionary single-objective optimisation, which allows us to capture experience from solving optimisation problems in form of empirical probability distributions as basis for model-based search operators. The statistical approach also allows us to define task similarity in terms of similarity of the retrieved distributions. We show the viability of this approach and discuss related issues on a selected variety of multi-modal problems. Again, we stress that we do not attempt to challenge existing model-based methods (e.g. [2], [3]), but stress that our school of thought can be seen as rather relating to recent research on transfer learning in optimisation [5]–[8]. In the latter, the perspective of the operators is seemingly missing. To our knowledge, aside from our own recent work [13], similar thinking has not been applied in the literature on continuous optimisation before.

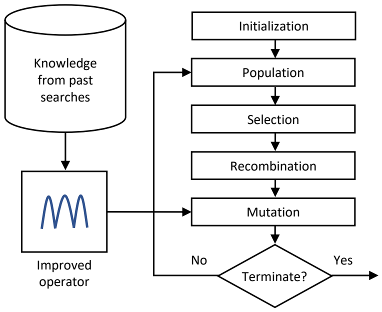

Fig. 1. Illustration of the framework we study in our experiments. From knowledge extracted from past searches on source tasks we build an improved operator which can be subsequently used to initialize an algorithm on a new target task.

III. Framework and Experiments

A. Algorithm and Setup

In our study, we consider continuous single-objective optimisation problems of the form $f : \chi \subseteq \mathbb{R}^d \rightarrow \mathbb{R}$, where χ is the search space and d its associated dimension. We further use as base the continuous genetic algorithm [14], which unlike its binary version does not distinguish between genotype and phenotype. Thus, solutions are directly represented in the search space by vectors

$$\mathbf{x}(j) = (x_1(j), x_2(j), \cdots, x_d(j)), \tag{1}$$

where the variable j simply indicates the j-th solution. Subsequently one can also define variation operators which act upon the solutions. In our following study we use the one point crossover operator defined analogously to the binary case [14] and draw mutations from a multivariate Gaussian mutation operator

$$\Delta\mathbf{x} \sim \mathcal{N}(\mathbf{x}|\,\mathbf{0}, \boldsymbol{\Sigma}), \tag{2}$$

with spherical covariance $\boldsymbol{\Sigma} = \mathbb{1} \cdot \sigma^{-2}$ and fixed sampling width σ, which upon mutation shifts solutions such that

$$\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}. \tag{3}$$

To foster the use of learned experience by means of improved operators, we keep in our framework track of mutations performed. The necessary modification to the genetic algorithm is illustrated in Fig.1. In principle, we only extend the standard architecture by a repository, which while in 'storing mode' is filled with copies of pairs of fitness values f and solution positions $\mathbf{x}$ from before and after application of the mutation

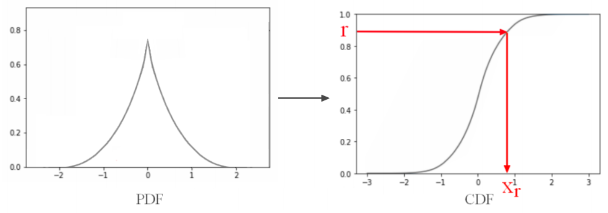

Fig. 2. Illustration of the inverse transform sampling technique [15]. The histogram to the left is used as a parametrization of an empirical probability density function (PDF). The latter is integrated to obtain a cumulative density function (CDF), which can be used to generate pseudo-random numbers x_r by inversion.

operator. This mutation tracking allows us in the following to further distinguish between *improving*

$$f(\mathbf{x}(j)^i_{\text{before}}) - f(\mathbf{x}(j)^i_{\text{after}}) \geq 0 \tag{4}$$

and *worsening mutations*

$$f(\mathbf{x}(j)^i_{\text{before}}) - f(\mathbf{x}(j)^i_{\text{after}}) < 0. \tag{5}$$

The idea is that once we have stored the mutations outside of the algorithm, we can filter them according to whether they are improving or worsening and subsequently aggregate them into bins to build histograms. The latter can be considered to serve as improved and problem-tailored search distributions for the problems of interest. In our case, we only harness histograms of improving mutations, as worsening mutations are strongly normal distributed [13]. Note, that the constructed histograms do not necessarily behave like Gaussian normal distributions, thus we have to explicitly use a resampling technique. For this reason we use a non-library implementation of the inverse transform sampling technique [15]. For a histogram with only one random variable we first calculate the cumulative density function given by

$$\text{CDF}(x) = \int_{-\infty}^{x} \rho(x')\, \mathrm{d}x'. \tag{6}$$

Note that $0 < \text{CDF}(x) < 1$, thus we uniformly sample a random number $r \in [0, 1]$ and use $\text{CDF}^{-1}(r)$ to generate a pseudo-random number x_r according to the distribution ρ (also c.f. Fig. 2). The multivariate case works analogously, however one starts first with a marginalized cumulative probability density and subsequently conditions it upon randomly generated components until a full point in the search space is obtained. Note, that in our study we use a bin size of 100 per dimension and choose the histogram widths h individually such that important features of the distributions are preserved.

Our experiments are mainly based upon a modified version of the DEAP library for evolutionary computation [17]. We choose the crossover probability to be 0.2, the mutation probability as 0.5, the population size as 30 and limit the maximum number of generations to 100. The sampling width of the mutation operator is, except when explicitly mentioned otherwise, set by default to $\sigma = 0.71$. Tournament selection with a size of 4 is further chosen. The start population is initialized randomly on the complete search space, where we keep the dimensionality fixed to $d = 2$ for all experiments. In all cases, except when explicitly mentioned, obtained minimum fitness values are averaged over 1000 runs to retrieve expressive statistics. We use in this paper the term fitness in the sense of a fitness cost which we want to minimize.

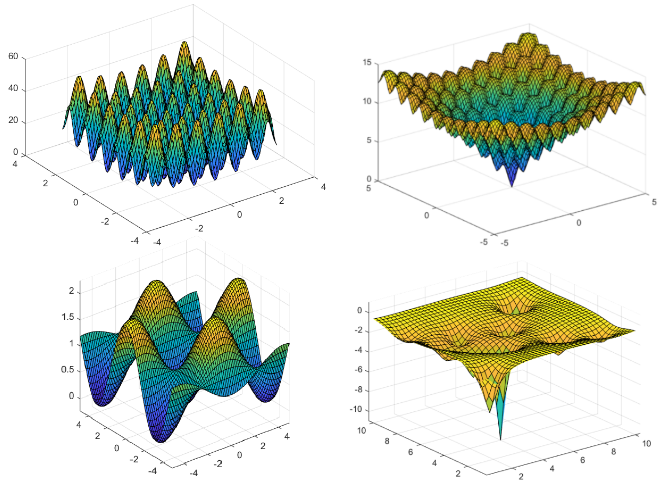

Fig. 3. Fitness landscapes of the benchmark functions considered within this study. Top row from left to right: Rastrigin's and Ackley's function. Bottom row: Griewank's and Shekel's function with the standard parameter setting commonly found in literature [16].

B. Experiments

1) Algorithm-problem interaction: We first start our 1st series of experiments with an investigation into the algorithm-problem interaction, as we expect that the balance in the interplay between both should lead to notable differences in the statistical distributions of improving mutations we can extract. For this reason, we run experiments using our extended genetic algorithm with the configuration as detailed in Sec. III-A. In the first series we keep the algorithm configuration constant and consider exclusively Ackley's benchmark function [14]

$$f(\mathbf{x}) = -a \exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=0}^{d} x_i^2}\right) + \exp\left(-\frac{1}{d}\sum_{i=0}^{d}\cos(2\pi x_i)\right) + a + \exp(1), \tag{7}$$

with $\chi = [-32.768, 32.768]^d$ and the depth parameter being usually defined as $a = 20$. However, in the following we vary a with the range from 20 to 1, thus varying the depth and steepness of the funnel while essentially keeping the positions of local extrema the same. In the 2nd series of experiments we consider Griewank's benchmark function. However, we keep the parameters constant and only vary the sampling width for mutations from $\sigma = 0.71$ to 4. The retrieved distributions from the first and second experiment series are illustrated in the first

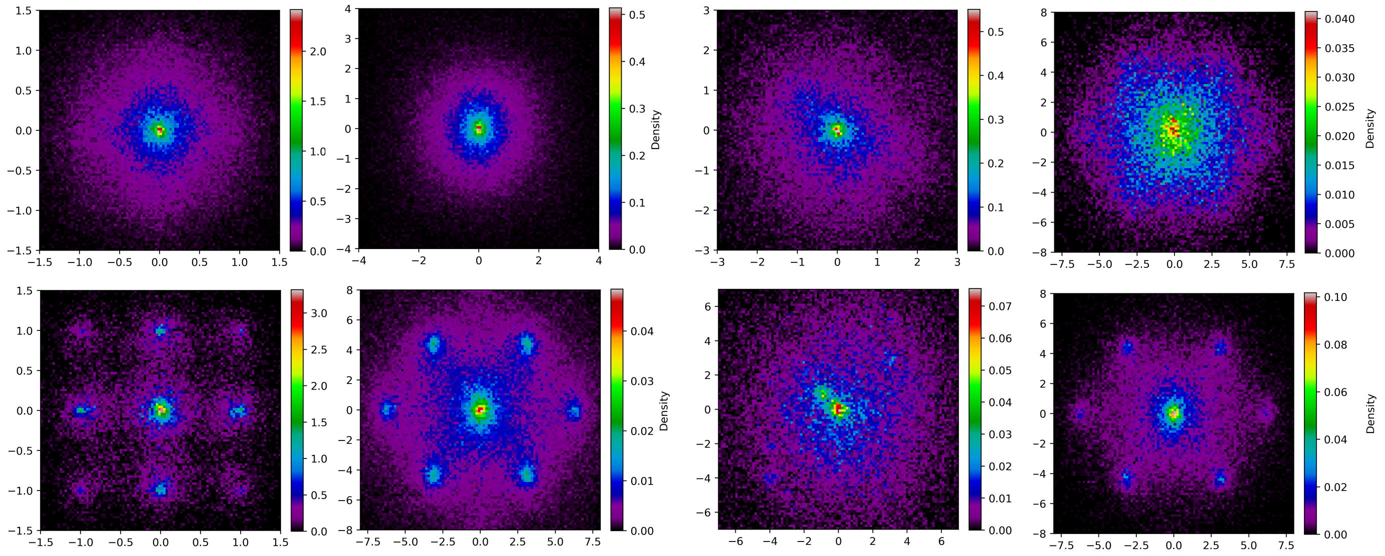

Fig. 4. *Column 1 and 2:* Demonstration on how the interplay between algorithm configuration and optimisation problem can effect the retrievable statistics. First column: Distributions retrieved on Ackley's function for $a = 20$ (top) and $a = 5$ (bottom) for unchanged algorithm configuration. Second column: Distributions obtained on Griewank's function for constant problem parameters and changed sampling width $\sigma = 0.71$ (top) and $\sigma = 4$ (bottom). *Column 3 and 4:* Retrieving distributions from multimodal problem and with irregular periodicity and investigation of the dynamics. Third column: Evolution of the distributions extracted on Griewank's function from generation 0 to 10 (top) and generation 10 to 40 (bottom). Intervals were chosen to achieve a same sampling quality of about $40,000$ samples. Fourth column: Extracted distribution on Shekel's function for $\sigma = 2$ (top) and $\sigma = 4$ (bottom).

and second column of Fig. 4.

We find that on Ackley's function for the parameter $a = 20$, the retrieved distribution resembles a simple multivariate normal distribution and does not seem to encode any problem specific information. Setting the parameter to $a = 1$, the retrieved distribution strongly differs from a Gaussian bell shape by having further peaks akin to grid points in a Moore neighborhood. The algorithm configuration thus is able to resolve notable problem-specific information. On Griewank's function [14]

$$f(\mathbf{x}) = 1 + \frac{1}{4000}\sum_{i=1}^{d} x_i^2 + \frac{1}{4000}\prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right), \qquad (8)$$

usually defined on the search space $\chi = [-600, 600]^d$ and without any further free parameters, we only vary the sampling width σ. We find that for $\sigma = 0.71$, the algorithm again only retrieves a Gaussian multivariate distribution. After significantly enhancing the sampling to $\sigma = 4$, we can however retrieve a neighborhood structure of peaks arranged in a hexagonal grid akin to the pattern in the fitness landscape in Fig. 3. Note, that we can interpret the recovered distribution as consisting out of a central part for local improvements and an outer part for long-range exploration of the neighborhood. As we considered so far only problems with a strong periodicity, we further investigate in the following the retrievable distributions from a problem with less obvious defined behaviour. For this reason we consider the generalized Shekel's benchmark function defined by [16]

$$f(\mathbf{x}) = -\sum_{i=1}^{m}\left(\sum_{j=1}^{d}(x_j - C_{ij})^2 + \beta_i\right), \qquad (9)$$

with search space $\chi = [0, 10]^d$, the number of dimensions d and the number of extrema m. The free parameters C_{ij} and β_i can in principle be set arbitrarily by the practitioner. We choose in our experiments standard settings found in the literature [16] for the parameters of the first two dimensions and include the usual ten extrema.

We find in Fig. 4 that for $\sigma = 2$ the density vaguely resembles a multivariate normal distribution, however includes distortions such that the region of high density has a flatiron shape. Enhancing the sampling width to $\sigma = 4$, the retrieved distribution picks up more peculiarities. In the inner high density regions an elongated double peak shape emerges along the diagonal, while in the outer regions smaller islands of increased density emerge. This reflects features in the fitness landscape of Shekel's function (c.f. Fig. 3).

The last series of experiments that we conduct briefly investigates the dynamics of the retrieved distribution. Particularly, as for the retrieval procedure we always assumed a full generational interval ranging from 0 to 100. For this reason, we run experiments on Griewank's function and aggregate mutations from the generational intervals 0 to 10 and 10 to 40. The intervals were chosen to approximately retrieve the same sample sizes of about $40,000$ improving mutations. We find that in the initial generations the distribution resembles to

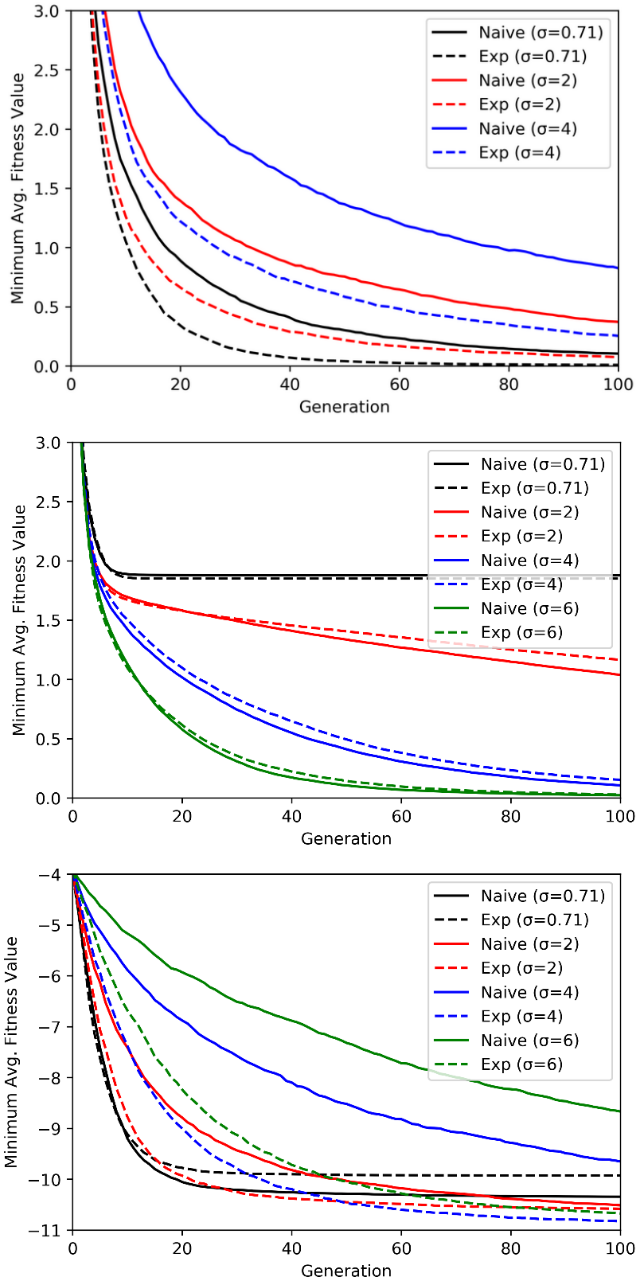

Fig. 5. *Top to bottom:* Comparison of the minimum fitness costs per generation averaged over 1000 runs for different sampling widths σ (continuous lines) and the resulting improved distributions (dashed lines) for Rastrigin's (top), Griewank's (center) and Shekel's (bottom) benchmark function.

a multivariate normal Gaussian. Only in later generations the prominent hexagonal grid structure emerges.

2) Sampling width and performance improvements: As we have seen in the previous series of experiments, that the sampling width is crucial in the retrieval of problem characteristic distributions, we investigate in the following the effect of it in regards to possible performance improvements. The resulting plots of average minimum fitness cost achieved over increasing generation are shown in Fig. 5. Again we have used the algorithm configuration as detailed in Sec. III-A. We find for Griewank's function that although it possesses many deceptive minima and although we can retrieve an improved distribution, increasing just the sampling width seems to be enough for performance improvements. In fact, for increased sampling width the retrieved distributions even slightly worsen the performance. For all tested settings, we could only retrieve for $\sigma = 0.71$ a better performing search distribution, however with statistically insignificant difference to the naive distribution when compared within a t-test (c.f. Tab. I). This seems to be counter-intuitive at first, however considering the fact that Griewank's function is characterized by a comparatively flat gradient and far apart extrema, reducing the sampling distribution to those of just improving mutations may impede the exploratory properties of the search too much to result in any performance improvements. Looking at Rastrigin's benchmark function

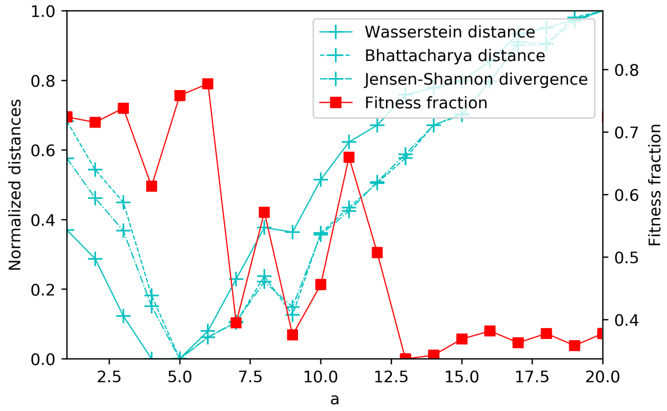

Fig. 6. Comparison of the behaviours of different statistical distance measures as possible proxies of task similarity to the fractional fitness cost achieved when transferring the improved sampling from Rastrigin's to Ackley's function. The depth parameter a of Ackley's benchmark function is varied. Note, that due to normalization 0 only indicates highest similarity and not sameness.

$$f(\mathbf{x}) = 10\,d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)] \tag{10}$$

with $\chi = [-5.12, 5.12]^d$ and Shekel's benchmark, we find that we can indeed retrieve distributions which can enhance the performance of the algorithm in a statistically significant manner. However, apart from Shekel's function, the distributions retrieved at higher sampling width on Rastrigin's function do not out-compete those for the smaller widths. Note, that we did not tested our procedure on Rastrigin for $\sigma = 6.0$, as this sampling width exceeds the size of the search space.

3) Test of statistical measures for task similarity: At last, an interesting question to investigate is whether or not the statistical distributions can be considered as proxies for task similarity in terms of improved performance, when transferring them to tasks of similar structure with similar retrievable

TABLE I
SAMPLING WIDTH σ, HISTOGRAM WIDTH h_{width}, BEST AVERAGE FITNESS VALUES $\overline{f}$, STANDARD DEVIATIONS $\overline{s}$ AND T-VALUES $t\text{-}val$ AFTER 100 GENERATIONS AVERAGED OVER 1000 RUNS. NOTE THAT $\overline{f}_{best}$ AND $\overline{s}_{best}$ STEM FROM THE NAIVE SAMPLING APPROACH, WHILE $\overline{f}'_{best}$ AND $\overline{s}'_{best}$ STEM FROM THE EXPERIENCE-BASED SAMPLING APPROACH. BEST ACHIEVED FITNESS VALUES WHERE THE NULL HYPOTHESIS FOR $\alpha = 0.05$ COULD BE REJECTED ARE MARKED IN BOLD FONT, WHILE AMBIGUOUS HIGH-PERFORMING VALUES ARE MERELY UNDERLINED.

	Rastrigin				**Griewank**				**Shekel**			
σ	0.71	2.00	4.00	6.00	0.71	2.00	4.00	6.00	0.71	2.00	4.00	6.00
h_{width}	1.5	3	5.5	-	1.5	6	11	15	1.5	3.5	6	8
$\overline{f}_{best}$	0.11	0.37	0.83	-	1.88	1.04	0.11	0.02	-10.34	-10.51	-9.48	-8.67
$\overline{s}_{best}$	0.19	0.37	0.62	-	2.16	1.18	0.13	0.02	1.84	1.08	1.51	1.96
$\overline{f}'_{best}$	**0.01**	**0.08**	**0.26**	-	<u>1.85</u>	1.17	0.15	0.03	**-9.93**	<u>-10.58</u>	**-10.83**	**-10.67**
$\overline{s}'_{best}$	0.01	0.12	0.30	-	2.13	1.33	0.19	0.03	2.30	1.47	0.52	0.69
$t\text{-}val$	16.17	23.90	26.35	-	0.30	2.26	6.28	4.93	4.47	1.18	23.30	30.48

distributions. For this reason, we consider Rastrigin's and Ackley's benchmark function. Both are characterized by the same positions of local extrema and the global optimum. However, the exponential function makes the neighborhoods around the extrema in Ackley's function significantly steeper. One could expect, that a beneficial distribution retrieved on Rastrigin's function therefore could also be beneficial for Ackley's function and effectively emulate the effects, when we would likewise extract a distribution from Ackley's. However, the similarity of both distributions should be the deciding factor for the effectiveness of such a procedure. For this reason, we investigate the impact on reducing the fitness when using an improved sampling procedure retrieved from optimizing Rastrigin's function by transferring it to Ackley's function. We vary the parameter a which controls the depth and steepness of the funnel and compare the behavior of the impact or fractional fitness of improved to standard sampling $f_{\text{exp}}/f_{\text{standard}}$ calculated from 1000 runs with the algorithm configuration from Sec. III-A, to the similarity of the retrieved distribution with the one from Rastrigin's function. We use as distance measures the Bhattacharya distance [18] given by

$$D_B(P,Q) = -\ln\left(\sum_{\mathbf{x}\in X}\sqrt{P(\mathbf{x})Q(\mathbf{x})}\right), \qquad (11)$$

where $\mathbf{x}$ corresponds to the center position of a bin, X being the set of all bins, P and Q to the two distributions which are to be compared: the Jensen-Shannon divergence [19] given by

$$\text{JSD}(P,Q) = \frac{1}{2}[D_{\text{KL}}(P\|M) + D_{\text{KL}}(Q\|M)], \qquad (12)$$

where D_{KL} is the Kullback-Leibler divergence and $M = 1/2(P+Q)$ an average mutual distribution; and at last the Wasserstein or Earth Mover's distance [20]

$$\text{EMD}(P,Q) = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}d_{i,j}\right) \Big/ \left(\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}\right), \qquad (13)$$

where d_{ij} are Euclidean distances between the bins of the distributions P and Q, and f_{ij} are flow coefficients which are calculated by solving the optimal transport problem [21].

Our considered case unfortunately proves to be a good counter example. While all distance measures in Fig. 6 are shown to behave similarly when varying the parameter a, the fractional fitness shows regularly occurring spikes which are not reflected in any of the distance measures. Although, one may attribute the latter to errors arising from the stochasticity of the search process, the overall global trend of the fractional fitness is of declining nature. This contrasts the behavior of the distance measures, which all show a clear change of sign in their first derivative near $a \approx 5$.

IV. CONCLUDING REMARKS

In conclusion, we investigated in our paper a transfer learning method for continuous single-objective optimisation based upon model-based mutation operators. For this reason, we tested the assumption that variable preferences arising in the algorithm-problem interaction can be used to design operators and compare tasks from an algorithm perspective. In our study, a continuous genetic algorithm with Gaussian mutation operator was used as a base from which we retrieved from optimisation runs distributions of improving mutations, performed an analysis on these and used them in the hope of realizing performance improvements on new problem instances.

Our observations show us, that retrieving a distribution which significantly differs from the default normal distribution relies crucially on the interplay between problem characteristics and algorithm configuration. Further, we also investigated the retrieval procedure on different optimisation problems. We have seen, that using the improved distribution works especially good on multimodal problems and surprisingly also on problems with low modality, however at times may hurt the careful balance of explorative and exploitative qualities of a search distribution. This especially seemed to have been the case on Griewank's function, where the performance gain was at best only minimal, but still not statistically significant.

We note that in our investigation we considered the integrated statistics over all generations as basis for our improved sampling, thus neglecting any dynamic components. However, on further inspection we find that only in later generations problem characteristic features are unveiled which are contrasting the default operator. At last, we investigated whether or not statistical measures can be considered as a measure of task similarity in our framework. While all tested measures have shown very similar behavior, relating them in our case to performance improvements has not shown significant correlation. However, we think that nevertheless the proposed concept of relating task similarity to statistical similarity is an interesting direction to further explore.

For our future work, we intend to extend the number of considered benchmark problems for our method such that we can draw clearer boundaries for its effectiveness. Further, we want to replace the search distribution approximation currently done through histograms by mixture-based density estimators. This could also allow us to include dynamic aspects through online reweighing of mixture components. The long term goal of our research can be considered as constructing a framework which enables us to learn operators that generalize over classes of continuous optimisation problems and subsequentially can be constructed for new unsolved black-box optimisation problems. Similar in the sense of generalizing from a set of training to test instances. While our research may closely relate to the current popular line of research in regards to algorithm selection in continuous optimisation, to the best of our knowledge a model-oriented perspective has not been taken as of yet. Similar, in the current co-evolutionary line of research on transfer learning in optimisation, the perspective of the operators is seemingly missing. Ideally, one would envision that the ideal transfer method would transfer both, optimized operators in combination with candidate solutions of high-fitness from prior optimisation tasks.

Acknowledgment

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 766186. It was also supported by the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531), and the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008).

References

[1] Edmund K. Burke, Michel Gendreau, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, and Rong Qu. Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society*, 64(12):1695–1724, 2013.

[2] Pedro Larrañaga and Jose A. Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer Science & Business Media, 2001.

[3] Nikolaus Hansen. The CMA evolution strategy: a comparing review. In I. Inza E. Bengoetxea J.A. Lozano, P. Larrañaga, editor, *Towards a New Evolutionary Computation*, pages 75–102. Springer, 2006.

[4] Donald R. Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[5] B. Da, A. Gupta, and Y. Ong. Curbing negative influences online for seamless transfer evolutionary optimization. *IEEE Transactions on Cybernetics*, no. 99:1–14, 2018.

[6] Liang Feng, Yew-Soon Ong, Siwei Jiang, and Abhishek Gupta. Autoencoding evolutionary search with learning across heterogeneous problems. *IEEE Transactions on Evolutionary Computation*, 21(5):760–772, 2017.

[7] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y. Ong, K. Tan, and A. K. Qin. Evolutionary multitasking via explicit autoencoding. *IEEE Transactions on Cybernetics*, 49(9):3457–3470, Sep. 2019.

[8] K. K. Bali, A. Gupta, L. Feng, Y. S. Ong, and Tan Puay Siew. Linearized domain adaptation in evolutionary multitasking. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1295–1302, June 2017.

[9] Min Jiang, Zhongqiang Huang, Liming Qiu, Wenzhen Huang, and Gary G Yen. Transfer learning-based dynamic multiobjective optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 22(4):501–514, 2017.

[10] Sushil J. Louis and John McDonnell. Learning with case-injected genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(4):316–328, 2004.

[11] Gan Ruan, Leandro L. Minku, Stefan Menzel, Bernhard Sendhoff, and Xin Yao. When and how to transfer knowledge in dynamic multi-objective optimization. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2034–2041. IEEE, 2019.

[12] A. Gupta, Y.S. Ong, B. Da, L. Feng, and S. Handoko. Measuring complementarity between function landscapes in evolutionary multitasking. In *2016 IEEE World Congress on Computational Intelligence*, 2016.

[13] S. Friess, P. Tiňo, S. Menzel, B. Sendhoff, and X. Yao. Learning transferable variation operators in a continuous genetic algorithm. In *2019 IEEE Symposium Series on Computational Intelligence*.

[14] Dan Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[15] L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag New York, 1st edition edition, 1986.

[16] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation*, 3(2):82–102, 1999.

[17] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, Jul 2012.

[18] Anil Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.

[19] Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, page 31. IEEE, 2004.

[20] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 59–66. IEEE, 1998.

[21] Rémi Flamary and Nicolas Courty. POT Python Optimal Transport library, 2017.