Learning Transferable Variation Operators in a Continuous Genetic Algorithm

Stephen Friess, Peter Tino, Stefan Menzel, Bernhard Sendhoff, Xin Yao

2019

Learning Transferable Variation Operators in a Continuous Genetic Algorithm

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

Learning Transferable Variation Operators in a Continuous Genetic Algorithm

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*—The notion of experience has often been neglected within the domain of evolutionary computation while in machine learning a large variety of methods has emerged in the recent years under the umbrella of transfer learning. Notably, realizing experience-based methods suffers from a variety of conceptual key problems. The first one being in regards to what constitutes problem-similarity from an algorithm perspective and the second one being what constitutes the transferable experience by itself. Ideally, one would envision that a learning optimization algorithm could be expected to act similarly to a human-problem solver who tackles novel tasks initially without any preconceptions. Experience only comes into play until sufficient similarity to known problems is established. Our paper therefore has two aims. First, to outline existing related fields and methodologies and highlight their insufficiencies. Second, to make the case for experience-based optimization by a demonstration using a novel and statistics-based approach with a real-coded genetic algorithm as a case study. In this paper we do not claim to construct universal problem solvers, but instead propose that from an algorithm-specific-view, problem characteristics can be learned and harnessed to improve future performance of similarly-structured optimization tasks.**

***Keywords*—Evolutionary Computation, Statistical Learning, Stochastic Optimization, Knowledge Transfer, Machine Learning**

I. Introduction

Historically, the school of thought concerning evolutionary computation as a means of problem solving may be traced back as early as to the late 1940s with Turing first proposing functionally similar mechanisms in his considerations on intelligent machines [1], [2]. The subsequent decades saw the advancement of these key ideas towards most of today's foundational framework. The rough concept of an evolutionary algorithm may therefore be described as follows [3]: Given a population of individuals within some environment that has limited resources, competition for these resources results in only the fittest individuals surviving, which in turn leads to a rise in fitness of the population as a whole. In mathematical terms, individuals of a population correspond to elements in a bounded set of candidate solutions which through an evolutionary process is sub-sequentially transformed using variation and selection operators such that the candidates become optimizers of a function which in turn emulates a computational mean to calculate fitness values.

Evolutionary algorithms have been proven as being viable for applications in structural engineering [3], with one frequently cited example being the design of a radio antenna for the ST5 spacecraft [4], [5], but also more recently for the training and evolution of deep neural network architectures [6]-[7]. However, advances in the development of evolutionary search methods have not stopped since the turn of the millennium. Notable progresses have been made towards multi-objective optimization [8], many-objective optimization [9] and statistics-based approaches [10]. The latter explicitly try to abandon the arbitrariness introduced by variation and selection operators. While evolutionary problem solving has been proven as immensely effective in studies, their effectiveness may be curbed by computationally expensive function evaluations in practical applications. For instance, the objective function evaluated at a single point in the search space might correspond to a value of aerodynamic performance obtained from a computational fluid dynamics simulation, which in turn may take from minutes up to several hours for a single calculation. For this reason, surrogate-assisted methods have been developed in engineering design, which try to spare function evaluations by operating on a regression model built prior or during to the optimization process [11], [12]. However, barely any of the existing work tries to consider to exploit similarities in the tasks self from an algorithm perspective. In this paper, we therefore propose a novel approach towards experience-based evolutionary computation.

We will first discuss in Section II existing works in the domain of continuous evolutionary optimization and outline their insufficiencies in regards to characterizing tasks and harnessing their similarities. Subsequently, in Section III we introduce the theoretical framework for our experiments. We note, that our key assumption is that what characterizes problems are not explicit landscape characteristics or similarities among solutions, but instead emerging preferences in stochastic properties of the variation and selection operators in the evolutionary search. Thus we assume that similar problems might be characterized by similar statistics they create during the search. Not only does this abstract notably from existing concepts of problem similarity, but also provides a means of defining an algorithm specific view of problems. We therefore investigate these ideas in a case-study using an extended continuous genetic algorithm. Finally, we conclude this study in Section IV and give an outlook on further interests of investigation. We remark, that in the spirit evolutionary computation, one likewise might find a suitable analogy in biology to our concept of problem similarity in the form of similar environmental pressures faced in the convergent evolution [13] of different species.

II. Related Work

The first notable step towards experience-based evolutionary optimization was made in 2004 through the CIGAR framework [14]. It proposes a cased-based approach where intermediate and final solutions from previously solved optimization tasks are kept in a storage. Whenever a new optimization problem is tackled, this case-base is queried and solutions are retrieved from similar previously tackled problems using a task-similarity measure. The latter are then used to partially initialize the population on the new task with them. The paper upfront remarks that defining task similarity measures is in principle non-trivial. As an alternative it suggests that at times it is more reasonable to operate on basis of solution similarity instead. As however, many different problems might have very similar intermediate solutions, suggests that as a way of coping with this uncertainty, the case-retrieval and subsequent injection procedure should be performed periodically during the optimization. We remark, that this procedure of periodical case-retrieval and solution injection is also reflected in many recently developed algorithms. Notable works to this regard 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 [15]-[17]. Note, that their work assumes that for effectiveness of their method, task similarity and thus complementarity can be or has been established a priori.

A more sophisticated approach utilizing a periodic injection procedure is represented by AMTEA [18]. Within their work Gaussian distributions are used to model the final populations from previously tackled optimization tasks. When new tasks are encountered, periodically a mixture model is constructed from the repository to approximate the current generation. The obtained weights of the mixture model are then used to sample proportionally new child solutions from the previously solved source tasks. Note, that their work reflects a problem similarity through solution similarity philosophy. However, otherwise barely any of the existing works have attempted to further characterize problem similarity. Most notable, at this point is the cumulative complementarity proposed in the context of MFEA [19]. However, the calculation of this measure is aside from benchmark functions for practical applications infeasible as it requires the explicit calculation of gradients and integrals. Note, that also the inclusion of gradient information is specific to the algorithm, as it performs local improvements in the sense of Lamarckian learning through hill climbing. A more agnostic measure proposed for task similarity is the Pearson correlation of ranked samples of source and target task, which has been used in the context of MFEA as computationally cheap alternative to calculating the cumulative complementarity [20]. However, notably might be too simplistic to consider any algorithm specific behaviour.

III. Framework and Experiments

A. Theoretical Framework

Similar to [14], we follow the definition of Mitchell [21] to define a machine-learning program. An algorithm is said to learn from experience E with respect to some class of source tasks S and performance measure P, if its performance at target tasks T, as measured by P, improves with experience E. Note that we took the liberty of extending the definition to differ between so called source tasks and target tasks. This differentiation is common within the research field of transfer learning [22], [23] and has been also used in recent literature on knowledge transfer in evolutionary computation [24]. From this definition we can derive the following use cases of interest: 1) $S = T$ the classical machine learning case where the source tasks are identical to the target tasks of interest, 2) $S \neq T$ the source tasks are different to the target tasks and at last 3) $S \sim T$ meaning source and target task possess some quantifiable notion of complementarity. E.g., the source tasks $S \subseteq T$ form a subset of the target tasks. The latter two cases are referred to as transfer learning in the literature.

We note that there is a gray area when trying to differ between case 2) and 3). Specifically, the notion of complementarity between two mathematically seemingly different problems might not be human-intuitive at all. On the other side, two seemingly complementary problems from a human perspective might not possess any forms of transferable experience from an algorithmic point of view. We therefore argue strongly for the algorithmic perspective: Two problems are complementary for an algorithm $\mathcal{A}$ if they possess beneficial transferable experience E in regards to each other.

B. Algorithm and Setup

1) Algorithm: In our study, we consider as a base the continuous genetic algorithm [25], [26]. Unlike the binary version, it does not differ between genotype and phenotype. Thus, solutions are directly represented in the search space by vectors

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

where n is the dimension of the search space χ and the variable 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 and draw mutations from a multivariate Gaussian mutation operator

$$\Delta\mathbf{x} \sim \mathcal{N} \cdot \exp[-\mathbf{x}^T \Sigma\ \mathbf{x}], \tag{2}$$

with diagonal covariance $\Sigma = \mathbb{1} \cdot \sigma^{-2}$ which upon mutation shifts solutions such that

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

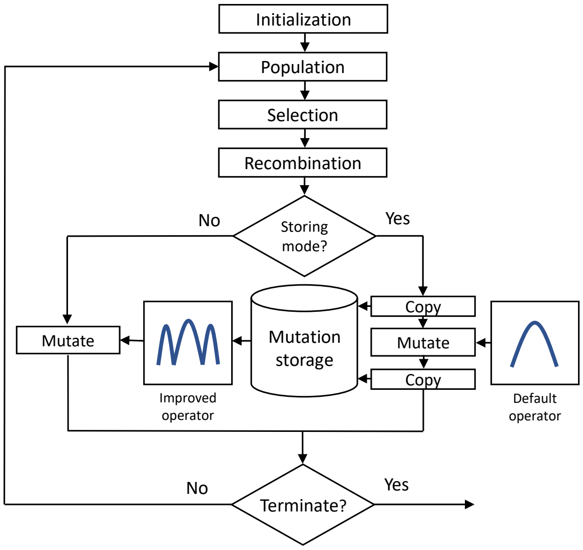

Fig. 1. Illustration of extensions we use within our algorithmic framework.

To foster the use of experience through assessment of problem characteristic properties in our framework, we keep in the following track of all mutations performed.

The necessary modification to the genetic algorithm is illustrated in Fig.1. We will further distinguish in the following between *improving*

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

and *worsening mutations*

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

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 B to build histograms. The latter can be considered to serve as better adapted distributions $\rho(x_1, x_2, \cdots, x_n)$ for mutation sampling on the problems of interest. 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 the inverse transform sampling technique [27]. For a histogram with only one random variable we first calculate the cumulative density function given by

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

Note that $0 < \mathrm{CDF}(x) < 1$, thus we uniformly sample a random number $u \in [0, 1]$ and use $\mathrm{CDF}^{-1}(u)$ to generate a pseudo-random number x_u according to the distribution ρ. 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.

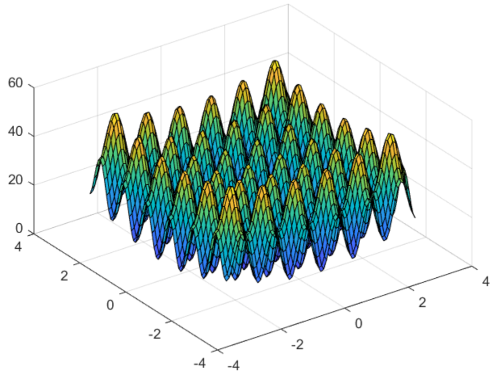

Fig. 2. Corresponding fitness landscape of Rastrigin's function. The benchmark problem is characterized by steep local extrema arranged in a grid pattern on top of a flat global gradient.

2) Setup: In the following we consider a series of experiments for demonstration on Rastrigin's function given by

$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)] \tag{7}$$

and illustrated in Fig.2. Additionally we will also consider Ackley's function which shares human-intuitive similarities to Rastrigin. Our experiments are based upon a modified version of the DEAP library for evolutionary computation [28]. 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 variance of the mutation operator is set to $\sigma = 0.71$. Tournament selection with a size of 4 is further chosen. The initial population is initialized randomly on the complete search space. In all cases, except when explicitly mentioned, obtained minimum fitness values are averaged over 1000 runs to retrieve expressive statistics. Note, that we use in this paper the term fitness in the sense of a fitness cost which we want to minimize.

C. Experiments

In the following, we first consider the scenario of learning and applying experience on the same original task. This experiment mainly serves as baseline to understand how

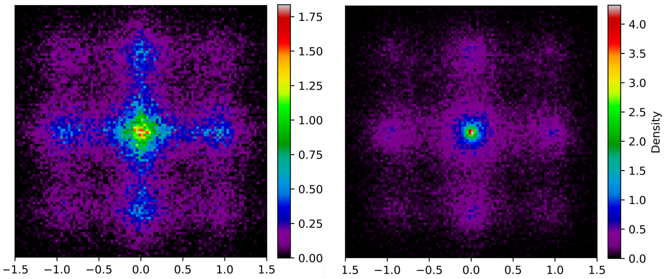

Fig. 3. Left panel: Sampling distribution modeled as independent in the random variables through usage of marginalized distributions. Right panel: Sampling distributions respecting the dependencies of the random variables.

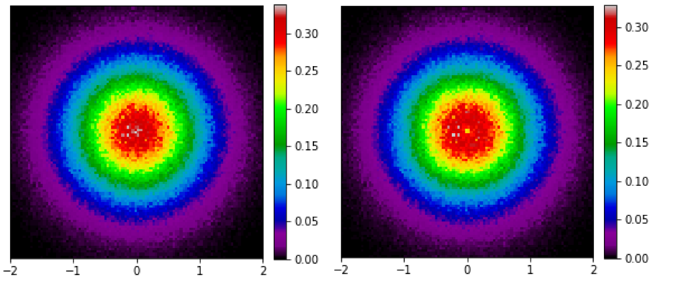

Fig. 4. Left panel: Gaussian bell shape of the full sampling distribution. Right panel: Sampling distribution of worsening mutations, visibly resembling strongly to a Gaussian except for the deviation in the origin.

and whether the concepted method is working correctly. Succeedingly, we consider the case of transferring learned experience to a task of different mathematical structure but with a similar structure from a human perspective. And at last, the case where the target task is a high dimensional generalization of the original one.

1) Identical source and target task: The problem of learning experience on a source task and applying to an identical target task mainly serves as a baseline. In order for one to conclude that viable experience E has been learned one would expect that first, experience can be harvested which significantly differs in its characteristics from the default stochastic behavior of the operators. And second, reapplying this experience to the 'training problem' shows significant performance improvement in comparison to the default approach.
In our case, we consider the distribution of improving mutations as the learnable experience we are interested in. For the case of a Gaussian variation operator, we expect that experience we want to harvest and reapply differs significantly from the standard statistical behavior of the variation operator.

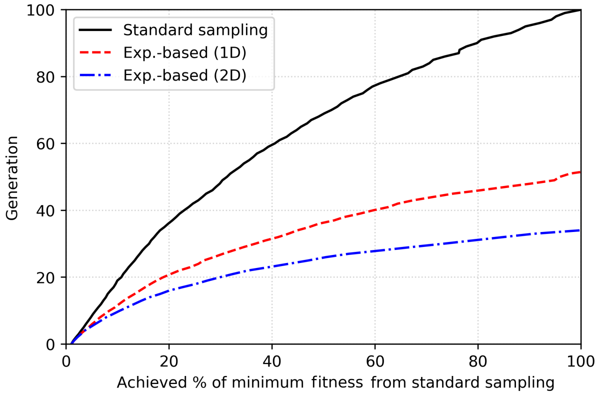

Fig. 5. Comparison of different experience-based sampling strategies to sampling variations without experience. The average minimum fitness at the origin corresponds to a value of $f = 9.87$, while the final achieved value of the naive run to $f = 0.10$.

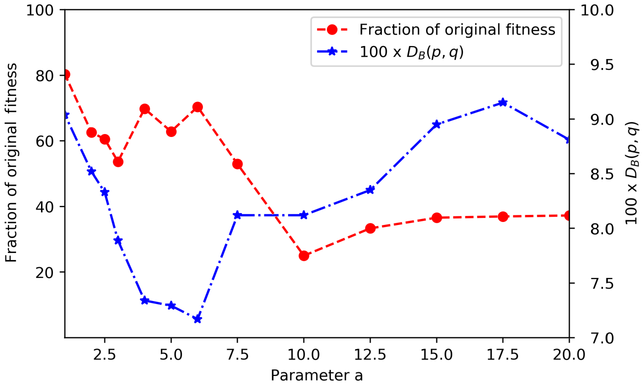

Fig. 6. Comparison of the behavior of the Bhattacharya distance and fraction of experience-based to naive fitness over the steepness parameter a after 100 generations. Notably, one finds that a relative reduction in the fitness does not translate necessarily into a reduction of the Bhattacharya distance.

We thus filter in the following for improving mutations and reconstruct the resulting distribution. It is shown in the right panel of Fig. 3 that the recovered distribution significantly differs from a Gaussian shape. For comparison, whereas in Fig. 4 the distribution of worsening mutations shows strong Gaussian behavior akin to that of the original variation operator. Thus one may conclude that the learned experience of 'worsening mutations' is not a helpful experience as it does not differ much from stochastic noise. In the following we therefore do not want to use this distribution to restrict the sampling.
In regards to the second expectation that reapplying the experience to the training problem should reveal an obvious performance improvement, we test this by explicitly sampling from the distribution of improving mutations. We further compare this to an approach where we model the sampling distribution as independent in its random variables $\rho(x, y) = \rho(x)\rho(y)$ by using marginalized distributions.
In the following we use the same algorithm configuration for our experiments as detailed in Section III-B2. The results are shown in Fig. 5, where we have chosen to plot the generation number over the percentage of achieved fitness in comparison to the value at generation 100 of the algorithm with standard sampling procedure, i.e. $x = 100\% \cdot (f_0^{\text{std}} - f_g)/(f_0^{\text{std}} - f_{100}^{\text{std}})$. We find significant performance improvement on the training problem, where by sampling from the new distribution we achieve the same fitness value as the 'naive' sampling approach after only 30% of function evaluations. Crudely approximating the experience-based sampling distribution as independent in the random variables we still achieve a competitive result at 50% of the function evaluations in comparison to the naive approach.

2) Different source and target task: In the following we consider the scenario of different source and target task. For this reason, we explicitly choose Ackley's functions, given by

$$f(\mathbf{x}) = -a\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=0}^{d} x_i^2}\right) + \exp\left(-\frac{1}{d}\sum_{i=0}^{d}\cos(2\pi x_i)\right) + a + \exp(1), \tag{8}$$

where the constants are usually chosen as $a = 20$, $b = 0.2$ and $c = 2\pi$. Ackley's function is characterized by a steep funnel towards the global optimum, steep gradients towards local minima, comparably flat outer regions and a large search space. However, it shares human-intuitive complementarities with Rastrigin since the location and frequency of local minima are partly the same. We therefore expect that if the 'experience' of optimizing Ackley's function is similar to that of Rastrigin's function, the experience gained from Rastrigin's function should likewise lead to performance gains on Ackley's function. To measure this 'experience' similarity we employ in the following the Bhattacharya distance [29]

$$D_B(p,q) = -\ln\left(\sum_{\mathbf{x}\in X}\sqrt{p(\mathbf{x})q(\mathbf{x})}\right), \tag{9}$$

where $\mathbf{x} \in X$ in our case corresponds to the vectors designating the respective bins. The Bhattacharya distance measures the degree of overlap between probability distributions. Note, that in principle other statistical measures such as the symmetrized Kullback-Leibler distance could also be used. However, we have chosen the former one as it handles better singularities appearing in discrete settings. In our experiments we again choose the same algorithm configuration as detailed in Section III-B2, but vary the steepness parameter a of Ackley's function over the experiments. We expect a 'less' steep function to be more akin to Rastrigin. The results are plotted in Fig. 6, where we compare the behavior of the Bhattacharya distance to the percentual fraction of average minimum fitness achieved after 100 generations in comparison to the achieved fitness without any transferred sampling procedure, i.e $y = 100\% \cdot f_{100}^{trans}/f_{100}^{std}$, over varying steepness parameter a. While an increasing similarity of Ackley's to Rastrigin's function is indeed reflected in the Bhattacharya distances at about $a = 5$, this is notably not translated into a performance increase for the experience-based sampling method.

Further, we test how sampling directly from the distribution of improving mutations built from Ackley's function compares to sampling from the distribution of Rastrigin's function. The corresponding result is shown in Fig.7. At first glance, it is evident that both improved sampling procedures work better on the target problem. However, their advantage is only realized after about 20 generations. Surprisingly this also holds true for the sampling distribution built from Ackley's function, which even performs slightly worse in comparison to the procedure from Rastrigin at generation 100. We attribute this behavior to the long tail evident after 20 generations in the naive sampling approach. Thus, improving mutations from generation 20 to 100 may simply be oversampled in comparison to the earlier beneficial ones.

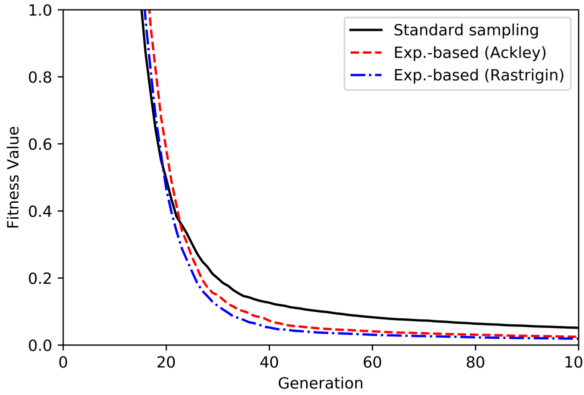

Fig. 7. Comparison of the minimum average fitness value achieved when using experience harvested on Ackley and Rastrigin to the naive approach.

3) Complementary source and target: Finally, we consider the scenario where a strong complementarity is given between source and target task. In our scenario, we consider higher dimensional generalizations of Rastrigins function and try to reapply the sampling distribution learned from the lower dimensional problem for $d = 2$. The source task is therefore a subproblem of the target task. In the following experiment we thus want to compare the fraction of fitness reduction $y = 100\% \cdot (f_{100}^{std} - f_{100}^{trans})/f_{100}^{std}$ we can achieve with rising dimension. We compare two approaches: In the first one, we again model the distribution through marginalized distributions and independent random variables, where in the second we explicitly use the 2d distribution which respects variable dependencies. While the former, is obviously more convenient for generalization to higher dimensions, the latter approach requires some additional considerations for use of

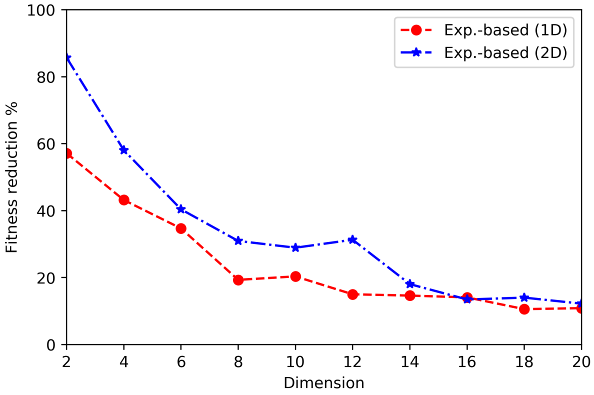

Fig. 8. Percentual fitness reduction after 100 generations on generalized Rastrigin's function for increasing dimension d through improved sampling procedures learned from $d = 2$. Compared are the improved samplings where the distribution is modeled through independent (red) and dependent (blue) random variables.

sampling in higher dimensional search spaces. In our case, we simply successively sample points from the 2d distribution and concatenate them to a vector until the latter has reached the dimension of the target task. To avoid introducing any biases by this partition, we additionally scramble the vector before applying it to mutate a solution in the population.

We again use the algorithm configuration as detailed in Section III-B2. The calculated results are plotted in Fig.8. As expected, the fitness reduction is the highest for the dimension of the source task. Overall, the sampling from the two-dimensional distribution achieves higher performance than from the crude uncorrelated approximation. However, most remarkable is that both methods still lead to a fitness reduction of up to $\approx 10\%$ upon ten-fold increase of dimensionality.

IV. Conclusion

In conclusion, the key contribution of our paper lies in making the point, that problems can be characterized by minable and learnable statistic properties. For this reason, we considered a case-study of a modified continuous genetic algorithm. As the modification allows us to store mutations of solutions, we can thus also filter them and build distributions of worsening and improving mutations. The distribution of worsening mutations has been shown not to differ statistically much from the default behavior of the mutation operator and was not considered further. However, the distribution of improving mutations encoded problem specific characteristics. Thus we reused this experience for improved sampling on our source problem and showed that on Rastrigin's function we could achieve competitive performance to the 'naive' approach after only 30% of the function evaluations. We could also show that transferring to the intuitive complementary Ackley's function likewise resulted in performance gains. However, the attempt to quantify this using the Bhattacharya distance did not turned out to be successful. Applying our resampling approach to Ackley's function reveals that it slightly performs worse than the experience from Rastrigin's function. We attributed this mismatch due to sampling on Ackley's function not being adequate in regards to the convergence characteristics. At last, we have tested how transferable the learned experience is to higher dimensional generalization of the Rastrigin benchmark function. When considering a ten-fold increase in the dimensions, we could still see a reduction of fitness of more than 10%.

For our future work, we plan to improve the distribution building process to account for any oversampling and possibly operate also on fewer samples. We may also further employ unsupervised machine learning to this regard as a computationally cheaper toolbox for resampling. The Bhattacharya distance has been tested as a means to quantify problem similarity, however has been shown to be inadequate for our demonstrated case. Thus it is of interest to find a proper distance metric. We note that our learning method is a form of statistics over fitness-gradient samples, where the integrated time evolution of the statistics is what we consider as the algorithm specific problem perspective. It is interesting to expand upon these considerations from a theoretical point of view. Further, it would be of interest to develop similar learning and transfer methods for state-of-the-art algorithms such as CMA-ES. At last, we note that the results for performance improvement upon the higher dimensional generalization are very motivating. As this could be useful for optimization problems where the search space dimension corresponds to an out of computational necessity chosen level of problem resolution. It would be a tremendous success if a similar strategy could be replicated and shown to save function evaluations on an expensive real-world optimization problem.

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 Peacock Plan (Grant No. KQTD2016112514355531), and the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008).

References

[1] A. M. Turing, "Intelligent machinery," Tech. Rep., 1948.

[2] A. M. Turing, "Computing Machinery and Intelligence," *Mind*, vol. LIX, no. 236, pp. 433–460, 10 1950.

[3] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer, 2003, vol. 53.

[4] G. Hornby, A. Globus, D. Linden, and J. Lohn, "Automated antenna design with evolutionary algorithms," in *Space 2006*, 2006, p. 7242.

[5] N. A. R. Center. Destination NASA - evolutionary antenna synthesis. https://www.nasa.gov/centers/ames/news/releases/2004/antenna/video.html.

[6] B. Wang, Y. Sun, B. Xue, and M. Zhang, "A hybrid ga-pso method for evolving architecture and short connections of deep convolutional neural networks," *arXiv preprint arXiv:1903.03893*, 2019.

[7] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[8] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *International conference on parallel problem solving from nature*. Springer, 2000, pp. 849–858.

[9] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[10] N. Hansen, "The CMA evolution strategy: a comparing review," in *Towards a New Evolutionary Computation*, J.A. Lozano, P. Larrañaga, I. Inza, E. Bengoetxea Eds., Springer, 2006, pp. 75–102.

[11] A. T. W. Min, Y. Ong, A. Gupta, and C. Goh, "Multiproblem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 15–28, Feb 2019.

[12] M. N. Le, Y. S. Ong, S. Menzel, Y. Jin, and B. Sendhoff, "Evolution by adapting surrogates," *Evolutionary Computation*, vol. 21, no. 2, pp. 313–340, 2013.

[13] J. B. Reece, L. A. Urry, M. L. Cain, S. A. Wasserman, P. V. Minorsky, R. B. Jackson *et al.*, *Campbell Biology*. Pearson Boston, 2016.

[14] S. J. Louis and J. McDonnell, "Learning with case-injected genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 4, pp. 316–328, 2004.

[15] L. Feng, Y.-S. Ong, S. Jiang, and A. Gupta, "Autoencoding evolutionary search with learning across heterogeneous problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 760–772, 2017.

[16] 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*, vol. 49, no. 9, pp. 3457–3470, Sep. 2019.

[17] K. K. Bali, A. Gupta, L. Feng, Y. S. Ong, and T. P. Siew, “Linearized domain adaptation in evolutionary multitasking,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1295–1302.

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

[19] A. Gupta, Y. 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.

[20] B. Da, Y.-S. Ong, L. Feng, A. K. Qin, A. Gupta, Z. Zhu, C.-K. Ting, K. Tang, and X. Yao, “Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results,” *arXiv preprint arXiv:1706.03470*, 2017.

[21] J. G. Carbonell, T. M. Mitchell, and R. S. Michalski, *Machine Learning: An Artificial Intelligence Approach*. Springer-Verlag, 1984.

[22] S. J. Pan, Q. Yang *et al.*, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[23] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.

[24] A. Gupta, Y.-S. Ong, and L. Feng, “Insights on transfer optimization: Because experience is the best teacher,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 51–64, 2017.

[25] D. Simon, *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[26] R. Chelouah and P. Siarry, “A continuous genetic algorithm designed for the global optimization of multimodal functions,” *Journal of Heuristics*, vol. 6, no. 2, pp. 191–213, Jun 2000.

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

[28] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, “DEAP: Evolutionary algorithms made easy,” *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, Jul 2012.

[29] A. Bhattacharyya, “On a measure of divergence between two statistical populations defined by their probability distributions,” *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.