To Appear In Evolution in Action: Past, Present and Future, Banzhaf et al. (Eds.), New York 2020. Springer.

Creative AI Through Evolutionary Computation

Risto Miikkulainen
The University of Texas at Austin
and Cognizant Technology Solutions

ABSTRACT

The main power of artificial intelligence is not in modeling what we already know, but in creating solutions that are new. Such solutions exist in extremely large, high-dimensional, and complex search spaces. Population-based search techniques, i.e. variants of evolutionary computation, are well suited to finding them. These techniques are also well positioned to take advantage of large-scale parallel computing resources, making creative AI through evolutionary computation the likely "next deep learning".

ARTICLE

In the last decade or so we have seen tremendous progress in Artificial Intelligence (AI). AI is now in the real world, powering applications that have a large practical impact. Most of it is based on modeling, i.e. machine learning of statistical models that make it possible to predict what the right decision might be in future situations. For example, we now have object recognition, speech recognition, game playing, language understanding, and machine translation systems that rival human performance, and in many cases exceed it [8, 9, 20]. In each of these cases, massive amounts of supervised data exists, specifying the right answer to each input case. With the massive amounts of computation that is now available, it is possible to train neural networks to take advantage of the data. Therefore, AI works great in tasks where we already know what needs to be done.

The next step for AI is machine creativity. Beyond modeling there is a large number of tasks where the correct, or even good, solutions are not known, but need to be discovered. For instance designing engineering solutions that perform well at low costs, or web pages that serve the users well, or even growth recipes for agriculture in controlled greenhouses are all tasks where human expertise is scarce and good solutions difficult to come by [4, 7, 11, 12, 18]. Methods for machine creativity have existed for decades. I believe we are now in a similar situation as deep learning was a few years ago: with the million-fold increase in computational power, those methods can now be used to scale up to real-world tasks.

Evolutionary computation is in a unique position to take advantage of that power, and become the next deep learning. To see why, let us consider how humans tackle a creative task, such as engineering design. A typical process starts with an existing design, perhaps an earlier one that needs to be improved or extended, or a design for a related task. The designer then makes changes to this solution and evaluates them. S/he keeps those changes that work well and discards those that do not, and iterates. It terminates when a desired level of performance is met, or when no better solutions can be found—at which point the process may be started again from a different initial solution. Such a process can be described as a hill-climbing process (Figure 1*a*). With good initial insight it is possible to find good solutions, but much of the space remains unexplored and many good solutions may be missed.

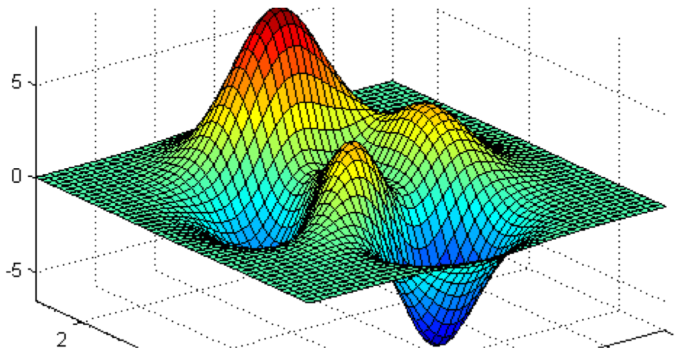
Interestingly, current machine learning methods are also based on hill-climbing. Neural networks and deep learning follow a gradient that is computed based on known examples of desired behavior [14, 22]. The gradient specifies how the neural network should be adjusted to make it perform slightly better, but it also does not have a global view of the landscape, i.e. where to start and which hill to climb. Similarly, reinforcement learning starts with an individual solution and then explores modifications around that solution, in order to estimate the gradient [21, 26]. With large enough networks and datasets and computing power, these methods have achieved remarkable successes in recent years.

However, the search landscape in creative tasks is likely to be less amenable to hill climbing (Figure 1*b*). There are three challenges: (1) The space is large, consisting of too many possible solutions to be explored fully, even with multiple restarts; (2) the space is high-dimensional, requiring that good values are found for many variables at once; and (3) the space is deceptive, consisting of multiple peaks and valleys, making it difficult to make progress through local search.

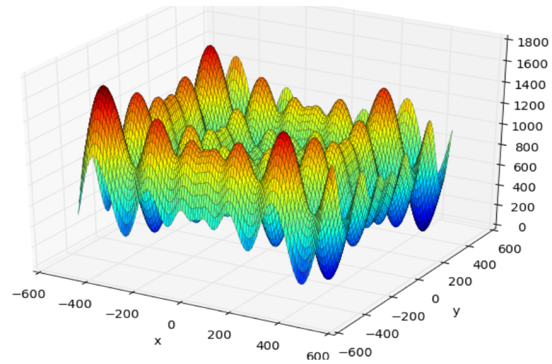
Evolutionary computation, as a population-based search technique, is in a unique position to meet these challenges. First, it makes it possible to explore many areas of the search space at once. In effect, evolution performs multiple parallel searches, not a single hill-climb. By itself such parallel search would result in only a linear improvement, however, the main advantage is that the searches interact: if there is a good partial solution found in one of the searches, the others can immediately take advantage of it as well. That is, evolution finds building blocks, or schemas, or stepping stones, that are then combined to form better comprehensive solutions [5, 10, 17].

This approach can be highly effective, as shown e.g. in the multiplexer benchmark problem [13]. Multiplexers are easy to design algorithmically: the task is to output the bit (among 2^n choices) specified by an n-bit address. However, as a search problem in the space of logical operations they grow very quickly, as $2^{2^{n+2^n}}$. There is, however, structure in that space that evolution can discover and utilize effectively. It turns out that evolution can discover solutions in extremely large such cases, including the 70-bit multiplexer (i.e. $n = 6$) with a search space of at least $2^{2^{70}}$ states. It is hard to conceptualize a number that large, but to give an idea, imagine having the number printed on a 10pt font on a piece of paper. It would take light 95 years to traverse from the beginning to the end of that number.

Second, population-based search makes it possible to find solutions in extremely high-dimensional search spaces as well. Whereas it is very difficult to build a model with high-order interactions beyond pairs or triples, the population represents such interactions



(a) Search Space Appropriate for Hill-Climbing



(a) Search Space in a Creative Domain

Figure 1: Challenge of Creative Problem Solving. Human design process as well as deep learning and reinforcement learning can be seen as hill-climbing processes. They work well as long as the search space is relatively small, low-dimensional, and well behaved. However, creative problems where solutions are not known may require search in a large, high-dimensional spaces with many local optima. Population-based search through evolutionary computation is well-suited for such problems: it discovers and utilizes partial solutions, searches along multiple objectives, and novelty. (Image credit: http://deap.readthedocs.io/en/latest/api/benchmarks.html)

implicitly, as the collection of actual combinations of values that exist in the good solutions in the population. Recombination of those solutions then makes it possible to collect good values for a large number of dimensions at once.

As an example, consider the problem of designing an optimal schedule for metal casting [3]. There are variables for number of each type of object to be made in each heat (i.e. melting process). The number of objects and heats can be grown from a few dozen, which can be solved with standard methods, to tens of thousands, resulting in billion variables. Yet, utilizing an initialization process and operators customized to exploit the structure in the problem, it is possible to find good combinations for them, i.e. find near-optimal solutions in a billion-dimensional space. Given that most search and optimization methods are limited to six orders of magnitude fewer variables, this scaleup makes it possible to apply optimization to entire new category of problems.

Third, population-based search can be adapted naturally to problems that are highly deceptive. One approach is to utilize multiple objectives [2]: if search gets stuck in one dimension, it is possible to make progress among other dimensions, and thereby get around deception. Another approach is to emphasize novelty, or diversity, of solutions in search [25]. The search does not simply try to maximize fitness, but also favors solutions that are different from those that already exist. Novelty can be expressed as part of fitness, or a separate objective, or serve as a minimum criterion for selection, or as a criterion for mate selection and survival [1, 6, 15, 16, 19].

For instance, in the composite novelty method [23], different objectives are defined for different aspects of performance, and combined so that they specify an area of search space with useful tradeoffs. Novelty is then used as the basis for selection and survival within this area. This method was illustrated in the problem of designing minimal sorting networks, which have to sort a set of *n* numbers correctly, but also consist of as few comparator elements as possible (which swap two numbers), and as few layers as possible (where comparisons can be performed in parallel). The search space is highly deceptive in that often the network structure needs to be changed substantially to make it smaller. Combining multiple objectives and novelty results in better solutions, and finds them faster, than traditional evolution, multiobjective evolution, and novelty search alone. The approach has already found a new minimal network for 20 inputs [24], and is now being extended to larger networks.

To conclude, evolutionary computation is an AI technology that is on the verge of a breakthrough, as a way to take machine creativity to the real world. Like deep learning, it can take advantage of the large amount of compute that is now becoming available. Because it is a population-based search method, it can scale with compute better than other machine learning approaches, which are largely based on hill-climbing. With evolution, we should see many applications in the near future where human creativity is augmented by evolutionary search in discovering complex solutions, such as those in engineering, healthcare, agriculture, financial technology, biotechnology, and e-commerce, resulting in more complex and more powerful solutions than are currently possible.

REFERENCES

[1] Cuccu, G., and Gomez, F. (2011). When novelty is not enough. In *Proceedings of the 2011 International Conference on Applications of Evolutionary Computation - Volume Part I*, 234–243. Berlin, Heidelberg: Springer-Verlag.

[2] Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. *PPSN VI*, 849–858.

[3] Deb, K., and Myburgh, C. (2017). A population-based fast algorithm for a billion-dimensional resource allocation problem with integer variablesbreaking the billion-variable barrier in real-world. *European Journal of Operational Research*, 261:460–474.

[4] Dupuis, J.-F., Fan, Z., and Goodman, E. (2015). Evolutionary design of discrete controllers for hybrid mechatronic systems. *International Journal of Systems Science*, 46:303–316.

[5] Forrest, S., and Mitchell, M. (1993). Relative building-block fitness and the building-block hypothesis. In Whitley, L. D., editor, *Foundations of Genetic Algorithms*, vol. 2, 109–126. Elsevier.

[6] Gomes, J., Mariano, P., and Christensen, A. L. (2015). Devising effective novelty search algorithms: A comprehensive empirical study. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 943–950. New York, NY: ACM.

[7] Harper, C. B., Johnson, A. J., Meyerson, E., Savas, T. L., and Miikkulainen, R. (2018). Flavor-cyber-agriculture: Optimization of plant metabolites in an open-source control environment through surrogate modeling. *bioRxiv*.

[8] Hassan Awadalla, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M., Liu, S., Liu, T.-Y., Luo, R., Menezes, A., Qin, T., Seide, F., Tan, X., Tian, F., Wu, L., Wu, S., Xia, Y., Zhang, D., Zhang, Z., and Zhou, M. (2018). Achieving human parity on automatic chinese to english news translation. Technical report, Microsoft Research.

[9] Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. G., and Silver, D. (2017). Rainbow: Combining improvements in deep reinforcement learning. *CoRR*, abs/1710.02298.

[10] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Ann Arbor, MI: University of Michigan Press.

[11] Hu, J., Goodman, E. D., Li, S., and Rosenberg, R. C. (2008). Automated synthesis of mechanical vibration absorbers using genetic programming. *Artificial Intelligence in Engineering Design and Manufacturing*, 22:207–217.

[12] Ishida Lab (2018). The n700 series shinkansen (bullet train). Retrieved 9/29/2018.

[13] Koza, J. R. (1991). A hierarchical approach to learning the boolean multiplexer function. In Rawlins, G. J. E., editor, *Foundations of Genetic Algorithms*, 171–192. Morgan Kaufmann.

[14] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–444.

[15] Lehman, J., and Stanley, K. O. (2010). Revising the evolutionary computation abstraction: Minimal criteria novelty search. In *Proceedings of the Genetic and Evolutionary Computation Conference*.

[16] McQuesten, P. (2002). *Cultural Enhancement of Neuroevolution*. PhD thesis, Department of Computer Sciences, The University of Texas at Austin, Austin, TX. Technical Report AI-02-295.

[17] Meyerson, E., and Miikkulainen, R. (2017). Discovering evolutionary stepping stones through behavior domination. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2017)*. Berlin, Germany.

[18] Miikkulainen, R., Iscoe, N., Shagrin, A., Rapp, R., Nazari, S., McGrath, P., Schoolland, C., Achkar, E., Brundage, M., Miller, J., Epstein, J., and Lamba, G. (2018). Sentient ascend: Ai-based massively multivariate conversion rate optimization. In *Proceedings of the Thirtieth Innovative Applications of Artificial Intelligence Conference*. AAAI.

[19] Mouret, J.-B., and Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary Computation*, 20:91–133.

[20] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Li, F. (2014). Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575.

[21] Salimans, T., Ho, J., Chen, X., and Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. *CoRR*, abs/1703.03864.

[22] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

[23] Shahrzad, H., Fink, D., and Miikkulainen, R. (2018). Enhanced optimization with composite objectives and novelty selection. In *Proceedings of the 2018 Conference on Artificial Life*. Tokyo, Japan.

[24] Shahrzad, H., Hodjat, B., Dolle, C., Denissov, A., Lau, S., Goodhew, D., Dyer, J., and Miikkulainen, R. (in press). Enhanced optimization with composite objectives and novelty pulsation. In *Genetic Programming Theory and Practice XVII*. New York: Springer.

[25] Stanley, K. O., and Lehman, J. (2015). *Why Greatness Cannot Be Planned: The Myth of the Objective*. Berlin: Springer.

[26] Zhang, X., Clune, J., and Stanley, K. O. (2017). On the relationship between the openai evolution strategy and stochastic gradient descent. *arXiv:1712.06564*.