Discovering the preference hypervolume: an interactive model for real world computational co-creativity

Hagg, A.

CHAPTER 6

In Conclusion

Creative processes, especially in engineering, suffer from the problem that user preferences are not easily formalizable. Creativity is suppressed when engineers are very experienced, causing them to become less inclined to try novel solutions. In contrast, a lack of experience makes it nearly impossible to find solutions at all but leaving the task of creation to machines takes away responsibility from us humans, which is not practical and raises many ethical questions.

Co-creative processes between humans and artificial intelligence can help to inspire beyond the user's own intuition. This work realized co-creative processes between humans and an artificial agent based on state of the art methods of evolutionary optimization and machine learning. By measuring similarity and diversity based on a solution's expression or behavior, multi-solution optimization creates a highly diverse set of solutions, both in their morphology as well as their interaction with the environment. The latter, whether it is a robotics or fluid dynamics simulation, determines not only whether a solution has a high quality but also offers alternatives when a quality metric is hard to specify. The co-creative system is able to create diverse solution sets in an efficient manner and capture human preferences.

Chapter 1 discussed that humans need to keep control of decisions made by AI systems. It grounded this thesis in the tension field between the Hegelian idea of creativity and engineering design. The communication paths of Jung's extraverted intuition, as opposed to introverted intuition, served as an inspiration. Increasing transparency by externalizing the creative process is one way to increase the explainability of decisions where ultimately AI systems are involved.

Chapter 2 analyzed formalisms of human creativity. The components of co-creative stage theories were connected to cognitive psychological traits, affecting human creativity. This analysis determined, which components of the process could be turned into an artificial system. The considerations, merged with those from literature, were converted to a list of requirements for the process. State of the art co-creative processes were then described. Finally, a formal model for a co-creative system based on divergent search methods was defined. The model supports humans with varying degrees of experience.

Chapter 3 connected ecologic effects in natural evolution to evolutionary multi-solution optimization. With the first research question (I) in this chapter, it was determined that solutions are best compared using their phenotype or behavior, not their genome. Niching methods that are common in evolutionary computation were discussed. Only by maintaining a collection of solutions can we differentiate between solutions' similarity. A description of genome, phenotype and behavior in natural evolution and their effects on diversity were discussed. These concepts were then projected back into the encodings used in evolutionary computation. Specifically, the terms phenotypic and ecologic expression were distinguished. The effects of neutrality and sensitivity of such multi-level encodings provided an answer to the research question. It is the phenotype, specifically morphology and behavior, that hold the key to a concept diversity that is easily understood by humans. QD algorithms fill a gap in evolutionary computation, by providing a mechanism for 'indirect, high-level recombination'. Values for phenotypic features, found in known solutions, are recombined in QD's archive to produce new solutions, but without the necessity of an explicit recombination operation.

Answering the second research question (II), it was determined that QD algorithms produce the highest phenotypic diversity compared to other multi-solution optimization methods. Three main multi-solution paradigms, multiobjective optimization, multimodal optimization and phenotypic niching (the main contribution of QD algorithms), were compared. A simplified archive was introduced that made it possible to compare genetic and phenotypic niching. QD was shown to create a much more diverse set of solutions. By not explicitly defining phenotypic features, and instead learning them from the data set generated by QD, solution diversity was increased even more. The treatment of the first research question already provided insight into why QD is such a powerful method and the answer to the second question confirmed this.

The next part in this chapter treated the question (III) whether we can learn more diverse solution sets when learning phenotypic niching from data instead of using predefined features. By taking advantage of convolutional autoencoders, phenotypic space could be compressed into two dimensions, increasing QD's solution set diversity. This removed the necessity for users to predefine phenotypic features, which can be difficult to determine as well as be prone to be influenced by a human creator's biases. This data-driven phenotypic niching approach, named Auto-Voronoi-Elites (AutoVE, see Fig. 3.15), has never been applied to shape optimization before. Indeed, data driven techniques lead to more diverse sets of artifacts.

The data-driven approach was further analyzed by asking what the limitations of generative models are in terms of the possible diversity of the solutions they create (research question IV). It turns out that generative models are better used as similarity models, providing niching spaces for multi-solution algorithms, rather than be used as an encoding. In this work only bootstrapped techniques were evaluated, to prevent initial biases and to simulate the problem of initial creativity. It was shown that generative models are usable to create a configurable resonance-dissonance trade-off, by evaluating their expressiveness when extrapolating away from known solutions. This extrapolation is only possible when using the models as niching space, not as search space.

Chapter 4 introduced two methods to help increase the data efficiency of QD algorithms. First, the question (V) was answered, whether behavioral features can be modeled in a surrogate-assisted way by sampling based on optimality alone. By connecting Bayesian optimization applied to QD with a model for phenotypic diversity, it was shown that the optimal hypersurface can be modeled, both in terms of its coordinates in phenotypic similarity as well as solutions' quality. Only now a full surrogate-assisted version of QD exists.

The second question (VI) treated a special case of neural encodings. An evaluation was performed to determine, whether the behavior of artifacts, robots in this case, with neural encodings can be modeled. It was shown that simulation-free behavioral prediction was possible by sampling outputs of neural networks and using this in a behavioral kernel for statistical models. This method removes the need for most evaluations of such an encoding and allow comparisons between solutions with different structures. The hope is that this helps to more easily choose such an encoding, which can create very complex morphologies and behaviors.

Chapter 5 discussed various ways to integrate QD into the co-creative process, using the design by shopping paradigm, where the user selects solutions presented to them. Results need to be presented to the user in a concise way, as QD can create very large solution sets.

First, the question (VII) whether QD results can be summarized using representatives was answered. Evidence that QD solution sets often consist of genetic clusters allowed compressing them into a smaller set of representative prototypes. An unsupervised clustering method combined with dimensionality reduction allowed such a compression of the set into a small number of dimensions. The set could now be presented to the user as prototypes.

Second, the prototypes were used to influence QD after humans select their preferred ones. This influence was analyzed for two domains in research question VIII. After the user selects their preferred solutions, QD is restarted from the set of selected prototypes. QD runs based on these 'seeds' produce solution sets that are more similar to the user's preferred prototypes. QD tends to drift away from the preferred region in genetic space.

Therefore, an evaluation was performed to determine, whether selected prototypical genomes can be modeled with a statistical model (IX), and whether we can constrain the search space by penalizing the objective function (X). Gaussian process models were created to predict the coordinates of new solution in the similarity space. This allowed defining a selection metric, which compares how far a solution has drifted from preferred solutions as opposed to non-preferred solutions. The metric was turned into a penalty and applied to QD's objective function. This biased QD to move towards the user's preference hypervolume, the space that contains preferred solutions. Two different encodings in the same domain were used to show that the drift penalty was able to constrain the solutions, genetically. However, due to the fact that the constraints are only applied to parameters, or genes, not to the phenotype, QD still produced unexpected solutions, especially when an encoding contains a high amount of neutrality and sensitivity.

Therefore, finally, the constraints were applied on a phenotypical level (XI). By modeling the high-dimensional space of pictures of geometrical shapes with a convolutional variational autoencoder, a phenotypic similarity space was created. The phenotypic drift penalty was applied to QD, creating the final co-creative process called HyperPref, in which the phenotypic preference hypervolume can be discovered. This was demonstrated on a 2D spline domain.

Introduced Methods. The following table shows an overview of all methods introduced in this work.

Table 6.1: Method Overview.

Method	Description
VE	Produces diverse set of high performing solutions using archive defined by phenotypic features, fixed number of niches, independent of archive dimensionality and without fixed niche boundaries.
AutoVE	VE adaptation that automatically extracts archive descriptors by bootstrapping and updating a generative phenotypic model.
SPHEN	Bayesian interpretation of MAP-Elites that uses GP models and a sampling strategy to fill an archive with only a small number of real objective evaluations. Predicts archive descriptors while piggybacking on sampling strategy.
PHD	Distance kernel/metric based on phenotypes.
UDHM	Preference model.
PRODUQD	Interactive version of QD. User influences QD by selection of archive members. Influence is imposed using a penalty applied to the objective based on genetic drift.
HyperPref	Version of PRODUQD that influences QD based on phenotypic drift.

Final Words This work gave insights into the matter of creativity in engineering, the importance of phenotypic diversity, reflecting on the formulation of optimization algorithms for the real world, bringing the user back into the loop, learning what the user wants, but providing as much creativity as possible within the user's selection. An informed and challenged engineer can make better design decisions and in this context we can and should develop algorithms.

Creators can use the Hegelian co-creative methods introduced in this thesis to discover their preferences, supported by a 'creative' machine. The machine helps humans interactively reflecting upon themselves as an objective participant, an artificial Other, a concept that is necessary, according to Sartre and Elkaïm-Sartre (1946). It helps to shape a process that is in line with the idea on creativity that was described by Hegel (1842) and uses communication paths used in extraverted intuition, the term coined by Jung (1923).

The formal model and the implementations of this model provide input for generative learning as well as the active learning community, combining concepts from both. The model can potentially support all levels of user expertise. The contributions of this work consist not only of answers to the research questions, but also of a body of methods that was shown on various peer reviewed conferences, in the community of evolutionary computation as well as computational creativity.

Surely, this work can be continued ad infinitum. The combination of QD and generative models is actively being explored in the scientific community. The aspects of interpolation and extrapolation w. r. t. known examples and features of those examples might provide us with insights on the true strength and limits of generative models. Although they produce convincing results, the question remains whether in the end they limit our possibilities. Only using data-driven solution generators might in the end hinder, not advance engineering solutions.

Cooperation between humans and artificial intelligence begs the question who or what bears the responsibility to take decisions that impact our society. Computational co-creativity allows us to evaluate these questions that can have a large impact on a society that depends more and more on AI, and less on human responsibilities. Yet, in the end, humans are the agents responsible for decisions (see Sartre and Elkaïm-Sartre (1946)).

To answer the question posed in Section 2.3.2, in contrast to the 'parameters tell the design story' hypothesis by Bradner et al. (2014), the results from this thesis quantitatively and qualitatively show that design stories, in terms of diversity, are told by examples (see Sartre and Elkaïm-Sartre (1946)) and their phenotypical features, not parameters. The objective of divergent search methods should be to provide new insights to the engineers. Parameters provide a computational search space but also contain implicit biases. Only a continuous analysis of phenotypes and behavior of solutions can provide new insights in an accessible manner.

I believe that in AI and optimization research, a larger emphasis should be put on the *process* of creativity and optimization. The optimization community oftentimes relies on the assumption that we can fully formalize our objectives and preferences. I do not believe this to be true. Instead, we might loosen this assumption and design methods, algorithms, and processes, that instead of giving us solutions, give us insight into the problems we try to solve and our preferences that grow with those insights.