Curious: Searching for Unknown Regions of Space with a Subpopulation-based Algorithm

Danilo Vasconcellos Vargas
Kyushu University
Fukuoka, Japan
vargas@cig.ees.kyushu-u.ac.jp

Junichi Murata
Kyushu University
Fukuoka, Japan
murata@cig.ees.kyushu-u.ac.jp

ABSTRACT

Intrinsic motivation and novelty search are promising approaches to deal with plateaus, deceptive functions and other exploration problems where using only the main objective function is insufficient. However, it is not clear until now how and if intrinsic motivation (novelty search) can improve single objective algorithms in general. The hurdle is that using multi-objective algorithms to deal with single-objective problems adds an unnecessary overhead such as the search for non-dominated solutions. Here, we propose the Curious algorithm which is the first multi-objective algorithm focused on solving single-objective problems. Curious uses two subpopulations algorithms. One subpopulation is dedicated for improving objective function values and another one is added to search for unknown regions of space based on objective prediction errors. By using a differential evolution operator, genes from individuals in all subpopulations are mixed. In this way, the promising regions (solutions with high fitness) and unknown regions (solutions with high prediction error) are searched simultaneously. Because of thus realized strong yet well controlled novelty search, the algorithm possesses powerful exploration ability and outperforms usual single-population based algorithms such as differential evolution. Thus, it demonstrates that the addition of intrinsic motivation is promising and should improve further single objective algorithms in general.

Keywords

Novelty Search, Learning-based Novelty Search, Differential Evolution, Prediction Error, Multiobjectivization, Intrinsic Motivation, General Subpopulation Framework

1. CURIOUS

Curious is a multi-objective algorithm for single-objective problems based on the general subpopulation framework [2]. The algorithm has two subpopulations. One for novel individuals called novel subpopulation and the other for the fittest individuals in the original objective function called main subpopulation. The choice of using two subpopulation instead of one single population is to avoid deleterious competition that could make one side (novelty or objective) stronger than the other. For a deeper discussion please refer to [2].

1.1 Initialization

Novel and main subpopulations are initialized to the same individuals, i.e., they are initially the same. Regarding the surrogate model, it is trained on a dataset with the same size of the main subpopulation. This dataset is made of randomly generated individuals, however, this dataset is created separately and therefore is entirely different from the main and novel subpopulations. In this first training stage the surrogate model learns until the mean squared error is smaller than an error threshold I_{et}.

1.2 Main Subpopulation

The main subpopulation behaves at least to a certain degree like a single-objective differential evolution (DE) [1]. One of the differences from DE is that the mutation operator selects individuals randomly from all subpopulations. The objective here is to let individuals from the novel subpopulation influence individuals in the main subpopulation, allowing the creation of mutation vectors close to novel or unknown regions of space. In fact, this creates a force to novel regions of space.

For each individual, DE's mutation is applied, with the exception that randomly selected individuals may come from any of the subpopulations.

Differently from DE's crossover, where there is only one way to create a trial vector, Curious has three ways:

- Usual crossover - The usual crossover is the same as the one used by DE;
- Novel crossover - Novel crossover is equation-wise the same as the usual crossover, however, the ith individual from the novel subpopulation n_i is used instead of parent x_i. This crossover is added to influence further the main subpopulation with novel subpopulation's individuals. The influence here is of a different type and degree, since becoming the parent of a crossover exerts extreme influence over the final trial vector (much more than with mutation vectors);
- Random individual - The trial vector becomes a randomly created individual. Random individuals are very important ingredients to construct surrogate models and that is why they are included here.

GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07...$15.00

DOI: http://dx.doi.org/10.1145/2908961.2908982

Crossovers have a chance of 50% of being a usual crossover, 25% of being a novel crossover and 25% chance of using a random individual as a trial vector. Basically, these percentages reflect a balance between two forces: the main subpopulation force of creating fitter solutions by perturbing the fittest solutions (50%) and the novel subpopulation force of creating novel solutions by perturbing novel solutions (25%) or just creating random ones (25%). The selection process is the same as DE's one, i.e., parent x_i is substituted by the trial vector u_i if and only if u_i is fitter than x_i.

1.3 Novel Subpopulation

The novel subpopulation behaves mostly like an archive, being updated every once in a while when a new individual (i.e., a trial vector u_i) has a novelty that is greater than one individual in the novel subpopulation.

In practical terms, the novel subpopulation has two main functions. One function is to propel solutions to unknown regions of the search space, for example, by influencing the creation of mutation or trial vectors. In this way, barely understood (prediction-wise) candidate solutions will be better investigated. This is based on the reasoning that novel solutions are present in unexplored or complex regions of space that deserve more attention of the optimization algorithm. The second function is to train the surrogate model in regions where it does not predict well. The assumption here is that the surrogate model will improve further by training where it does not predict well as well as will not forget to some degree what it learned previously. That is why choosing a good surrogate model is important and comparison between surrogate models is an important research point, however, it goes out of the scope of this article.

1.4 Surrogate Model

Every generation the surrogate model is used to evaluate the novelty of individuals. In the end of every generation, the surrogate model is updated using the novel subpopulation as training dataset. However, this time the learning occurs for only U_s steps and stops. This allows the model to improve somewhat its accuracy but avoid both overfitting and fine tuning. Recall that fine tuning is also unnecessary for surrogate models in novelty search, because surrogate models here need to be able to precisely compare candidate solutions instead of precisely evaluating one.

2. EXPERIMENTS

In this Section the objective is to verify the benefits or demerits of adding an intrinsic motivation (novelty search) to an optimization algorithm. Curious is similar to DE in many ways and although it employs a surrogate model, the surrogate model is used only for calculating novelty (i.e., the surrogate model was not used to reduce function evaluations or substituting the objective function in any form). Therefore it seems plausible to compare Curious to DE.

Experiments were conducted for both DE and Curious in all noiseless 24 problems of the BBOB-2015 benchmark.

2.1 Settings

The parameters for Curious is presented in Table 1. DE uses the same parameters, i.e., same CR, F and population size. All tests had a maximum number of function evaluations set to $10^5 \cdot dim$, where dim is the number of dimensions of the problem. Since the search domain for BBOB functions is $[-5, 5]^D$, all variables were also limited inside the $[-5, 5]$ range.

Table 1: Curious's Parameters

Main Subpopulation size	100
Novel Subpopulation size	100
CR	0.2
F	0.1
Surrogate Model	
Type	Neural Network
Hidden layer	1
Hidden nodes	$5 \cdot dim$
Hidden nodes' type	Logistic
Output nodes' type	Linear
Training algorithm	Levenberg-Marquardt
Error function	Mean Squared Error
Initial error threshold (I_{et})	10^{-5}
Updating steps (U_s)	30

Table 2: Mean Results in two dimensions ($dim = 2$). Δf_{DE} and $\Delta f_{Curious}$ are the mean difference between the optimum fitness and the best fitness found by respectively DE and Curious algorithms. The reported results were averaged over 15 instances of the same problem. Values below 10^{-4} are considered 0. Functions where there was no difference in performance between algorithms were omitted.

Function	Δf_{DE}	$\Delta f_{Curious}$	$\Delta f_{DE} - \Delta f_{Curious}$
f12	0.829	0	0.829
f15	0.066	0	0.066
f21	0.278	0	0.278
f22	0.513	0	0.513
f23	0.108	0.101	0.007
f24	0.038	0.009	0.028

2.2 Results

Table 2 shows the results for $dim = 2$ (i.e., 2 dimensions). Curious surpassed DE in six out of the 24 problems and behaved similarly in the remaining ones. This result alone justifies the employment of intrinsic motivation in single objective optimization.

It is possible to see that Curious has, in comparison with DE, relatively less difficulties with functions (problems) that continuously change directions (f12) and multi-modal functions with weak structure (f21, f22, f23 and f24). However, tests in higher dimensions are needed to study deeper the effects.

3. REFERENCES

[1] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces.

[2] D. V. Vargas, J. Murata, H. Takano, and A. C. B. Delbem. General subpopulation framework and taming the conflict inside populations. *Evolutionary computation*, 23(1):1–36, 2015.