New Inflationary Potentials with Evolutionary Algorithms

Chun Hin Ma

Abstract

This paper presents a novel genetic programming bottom-up approach to identify new inflationary potentials through the crossover between the tree representations of the function classes of natural inflation and arctan inflation. The resulting function class, defined as $\text{ats}_{(c_1,c_2,c_3)}(\phi) \equiv c_1 * \arctan(c_2 * \sin(c_3 * \phi))$, contains viable potentials within certain coefficients constraints. This is found by defining criteria to evaluate the viability of functions based on their cosmological predictions. This constraint the values of the normalization and scaling constants the function class could take, and serve as a probe of the topology of viable regions in the parameter space. Additionally, a mutation mechanism of function class has been introduced by using symbolic regression to discover a new inflationary function class $c_1 * |\arctan(c_2 * \phi)|^{c_3}$, which can be viable if appropriate constants constraints are satisfied. The findings demonstrate the effectiveness of genetic programming and symbolic regression in uncovering new inflationary potentials from non-trivial function classes.

1 Introduction

Cosmic inflation is a proposed solution to several cosmological puzzles, including the flatness, horizon, and monopole problem [1], [2]. The flatness problem refers to the near-perfect flatness of the universe's curvature, the horizon problem refers to the high degree of isotropy of the cosmic microwave background radiation, and the monopole problem refers to the absence of evidence of observable magnetic monopoles. Inflation also explains the origin of large-scale structures in the universe, which are generated by primordial inhomogeneities in the very early universe under quantum fluctuations. According to the inflationary model, the universe underwent an exponential expansion after the Planck epoch about 10^{-36} seconds after the Big Bang singularity, which smoothed out the curvature of the universe, resulting in its observed flatness, and resolved the uniformity of temperature across distant regions that could not have enough time to interact. In addition, inflation dilutes the density of magnetic monopoles, explaining their absence in current observations. Although cosmic inflation remains a speculative idea, it is supported by the power spectrum of the cosmic microwave background temperature anisotropies and serves as a connection between cosmology and string theory to the field of string cosmology. However, direct evidence of inflation would require the detection of primordial gravitational waves, which have not been found yet.

Modern inflationary theories often rely on the slow roll of a scalar field ϕ, known as the inflaton. In the slow-roll inflation scheme, the inflaton rolls slowly such that the time derivative of the inflaton is much smaller than the energy scale of the effective inflationary potential, i.e. $\dot{\phi} \ll |V(\phi)|$. Assuming the potential $V(\phi)$ is spatially homogeneous, its energy density and pressure are related by

$$\rho = \frac{1}{2}\dot{\phi}^2 + V(\phi), \quad p = \frac{1}{2}\dot{\phi}^2 - V(\phi) \tag{1}$$

The equation of state becomes $\rho = -p$, which depends on the inflationary potential $V(\phi)$. The exact expression for the inflationary potential $V(\phi)$ is unknown, but approaches to describe inflation could be broadly divided into either a top-down or bottom-up approach [1]. While a top-down approach derives V from a more fundamental parent theory, a bottom-up phenomenological approach works with an effective field theory (EFT). In terms of an EFT, as long as the inflationary function produces cosmological predictions in accordance with our observation, $V(\phi)$ is labeled as viable since it poses a candidate for the inflationary potential.

This paper adopts a bottom-up approach to identify viable functions for the inflationary potentials. This work is inspired by the previous studies on integrating cosmic inflation with genetic algorithms [3], [4], and results from *Encyclopædia Inflationaris* [5]. Ref [3] shows that Genetic Algorithms (GAs) could be used to search for desirable solutions more effectively than an exhaustive search in the context of cosmic inflation. A fitness function is a functional $F : \{f\} \to \mathbb{R}$ that maps a real single-valued function $\{f\} = V(\phi)$ to a real number $F\{f\}$. This functional is termed the fitness function and defines the landscape of this discrete optimization problem. The population with maximum fitness is of interest because they are candidates for the inflationary potential. As an optimization problem, GA has been shown to be an effective strategy to explore the search space to find desired candidates. GA is a *metaheuristic* that belongs to the class of evolutionary algorithms (EAs), which involves a sequence of instructions that mimic biological evolutions, such as selection, reproduction, and mutation. In this paper, a variant called Genetic Programming (GP) has also been employed to identify viable potentials by the 'breeding' of two function classes that are shown to be viable [5], demonstrating evolutionary algorithms are excellent in tackling the described task above.

Regarding the structure of the paper, in Sec. II, the viability of a function is quantified by discussing the fitness function used to compute the cosmological observables the function produces. In Sec. III, concepts of evolutionary algorithms,

specifically genetic algorithms and genetic programming, are reviewed. In Sec. IV, genetic programming is demonstrated as a tool to identify new inflationary potentials, and the results are analyzed accordingly. In Sec. V, a mutation mechanism using symbolic regression on function classes is shown to be able to identify new inflationary potentials. Finally, in Sec. VI, the conclusions are summarized.

2 Fitness Function

As mentioned in the introduction, the goal of this section is to quantify a given function's viability within the slow-roll framework. Most functions are not viable because they either fail to fulfill the assumptions for slow-roll dynamics or they do not produce predictions that match our observations. For example, consider an infinite step function, as the inflaton rolls towards the step in this model, not only does the function violate slow-roll dynamics, it also produces an infinite amount of radiation during the reheating period, a situation we do not observe in this universe. On the other hand, a constant function does not allow for spontaneous symmetry breaking and results in hardly any rolling of the inflaton. Therefore, we expect that viable functions are smooth functions and exhibit some slope without being too steep.

For a given inflationary potential, if a region of slow-roll could be identified, at least 4 cosmological predictions could then be computed: the number of e-folds the universe expanded, the spectral index n_s for the power law for the scalar perturbations, the tensor-to-scalar perturbation ratio r_*, and the energy scale of inflation m_*. Constraints are imposed on these four predictions to determine if a potential is viable. Specifically, it is required the total number of e-folds should be larger than the reference value which is about 60 to resolve the horizon problem. The predicted value for n_s at the pivot scale $k_* = 0.05 Mpc^{-1}$ has to be within $n_s = 0.9649 \pm 0.0042$, and r_* should be bounded by $r < 0.061$ from the Planck 2018 results [6]. The energy scale of inflation is fixed by the normalization of the present-day power spectrum of density perturbations $(V_*/\epsilon_*)^{1/4} = 6.7 \cdot 10^{16}$ GeV with a theoretical uncertainty of about 10% [7].

To quantify the viability of a function, We construct a fitness function that maps a function to a real value smaller or equal to zero. A potential is viable only if it has fitness zero, and the more negative the fitness, the less likely the potential is to be viable under the slow-roll approximations. The fitness function considers the four cosmological prediction contributions mentioned above. Working under the Planck units, and a search interval of $\phi = [-30, 30]$, redefine the input function such that the 'global' minimum within the search interval is centered at $\phi = 0$ with $V(\phi = 0) = 0$, such that $V(\phi) \geq 0$. Introduce two slow-roll parameters ϵ and η defined by

$$\epsilon = \frac{1}{2}\left(\frac{V'}{V}\right)^2 < 0.5, \quad \eta = \left|\frac{V''}{V}\right| < 0.5 \tag{2}$$

When the slow-roll parameters are within this approximation, this implies two 'flatness' conditions are met, which can be used to identify the path of slow-roll [8]. A reverse-engineering approach to finding the slow-roll path is as followed: the algorithm iterates with a step size $\Delta\phi = 0.006$ in both directions at the center to search for the first ϕ that satisfies (2), marking the end of inflation because the slow-roll approximation is about to not hold. The regions between the end and the trough could be the period of reheating. The starting point of inflation is found by continue moving up the potential. While keeping the same criteria for the slow-roll parameters, an extra condition has been applied by requiring the amplitude of quantum fluctuations of the inflaton field to be much smaller than the change in the classical amplitude of the field during a time interval of H^{-1}. This is represented by the inequality [3]

$$\delta\phi_q \approx \frac{\sqrt{V/3}}{2\pi} \ll 2\pi \frac{|V'|}{3V} = \delta\phi_c \tag{3}$$

In this study, choose

$$\delta\phi_q < 0.5 \cdot \delta\phi_c \tag{4}$$

The value of ϕ at which equations 2 or 4 fail to hold is taken as the initial point of inflation. If the conditions continue to hold up to the boundary, then the initial point is taken as the field value at the boundary. The region beyond the starting point of inflation is considered eternal inflation, and is not the focus of this study. Denote ϕ_e and ϕ_i as the ending and starting points of the inflation respectively, they can then be used to compute the four mentioned cosmological observables. Based on how far away the predictions are from current observations, the 'fitness' of this function can be quantified by considering the four contributions. The detailed implementation is included in the appendix. Thus, this fitness function defines a fitness value for every possible single-valued function when a slow-roll region could be identified.

Functions under this specific fitness landscape can be broadly classified into 3 types. The first type is functions that are discontinuous. The second type is those when an ending point of inflation could not be located within the interval probed on either side which made the observables ill-defined. The third type of function contains at least an ending point of inflation such that its fitness is well-defined. For the functions encountered in this paper, the typical fitness value of the third type

ranges from about -10 to 0. To penalize heavily on the first two types of function, the fitness for the first type has been set at -30 and the second type has been set at -25. The motivation is to accelerate the rate for the population to evolve away from the regions where slow-roll approximations could not be valid during the evolutionary algorithms.

3 Evolutionary Algorithms

Evolutionary Algorithm (EA) is a heuristic method for searching problems, which draws inspiration from biological processes such as natural selection. A typical EA involves two key components: a fitness function that evaluates the performance of a candidate, and a reproduction mechanism that generates new individuals by combining the 'genetic material' of the selected parents. These individuals may be represented in various forms, such as DNA, binary chains, or symbolic expressions. Once the two key components are established, the EA can follow a standard procedure that includes initialization, reproduction/evolution epochs, and termination.

At the start of the algorithm, a population of N_{pop} individuals is initialized, and the fitness of each is evaluated and ranked in descending order. The population is then allowed to reproduce according to the defined mechanism. A popular implementation of this mechanism is to allow individuals with higher fitness to have a higher chance of reproducing. After generating N_{pop} new individuals in a generation, the resulting population serves as the base population for the next generation. This cycle repeats for several generations until the algorithm terminates after some criteria have been met. Different termination criteria could be used, but one has to prevent the algorithm from being stuck in an infinite loop. Overall, EA is a versatile method that can be applied to various types of problems. By appropriately setting up the two key components, EA could be used to identify inflationary potentials.

3.1 Genetic Algorithms

Genetic Algorithms (GAs) belong to the class of EAs and share the same main structure as described earlier. As shown in Ref [3], GAs are effective in identifying viable regions within the coefficient space of a given function class. This is achieved by encoding the coefficients of a function class into a binary chain, where each chain has its own fitness value, and a breeding mechanism can be easily defined by the manipulation of these chains, forming the basis of an EA.

By specifying the GA to work on a selected parameter space, it can be chopped into discrete sub-regions, with each region labeled by distinct binary chains. These chains are analogous to chromosomes and their components are analogous to genes, hence the name 'genetic algorithm'. The benefit of this encoding is that the strings can be easily manipulated during the reproduction process. Offspring can be produced from two parent strings by swapping their components, and each gene has some probability to mutate to a different configuration. In this study, GAs have been employed on a new function class by a similar encoding to identify viable potentials within the new function class's parameter space, as described in Sec 4.2.

3.2 Genetic Programming

Genetic programming (GP) is a variant of genetic algorithms and is used to evolve mathematical expressions. Each individual in the population is now a function class, and mathematical expressions could be expressed in a tree form. For example,

$$c_1 + c_2 * \sin(c_3 * \phi) \longleftrightarrow [\ +\ ,\ c_1\ ,\ *\ ,\ c_2\ ,\ \sin\ ,\ *\ ,\ c_3\ ,\ \phi\ ] \tag{5}$$

$$c_1 + c_2 * \arctan(c_3 * \phi) \longleftrightarrow [\ +\ ,\ c_1\ ,\ *\ ,\ c_2\ ,\ \arctan\ ,\ *\ ,\ c_3\ ,\ \phi\ ] \tag{6}$$

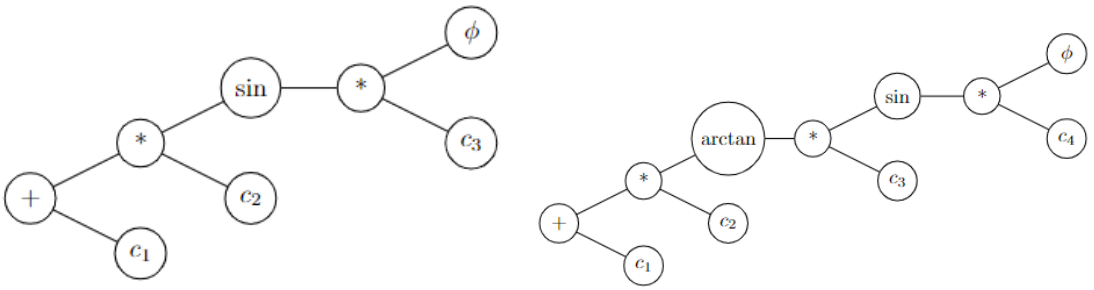
Three types of operators are considered, which take different numbers of arguments. A binary operator (e.g. + , x , pow) takes two arguments, which act as a 'parent' node as it carries two 'child' nodes as its arguments in order to generate a valid tree. A unary operator (e.g. exp, log) takes a single argument and carries one 'child' node. A nullary operator (e.g. constants, parameters) takes no argument and terminates that path of the tree. This defines the rules for traversing the tree to recover the symbolic expression, and the length of the list corresponds to the 'complexity' of that expression.

Upon identifying the individuals in the context of GP, the two key components of EA has to be taken care of. Since we are working on the task of finding viable potentials, a measure of fitness could be the normalized n-volume of viable potentials within the parameter space of the function class, where n is the number of coefficients of the function class. Intuitively, for a very viable function class, it should contain large regions of viable potentials within its parameter space. For a function class to be not very viable, it should contain few to no viable regions within its parameter space.

After justifying the existence of a fitness measure on function classes, the reproduction mechanism of function classes has to be defined. This can be achieved by first selecting two parent trees for breeding, then picking a valid sub-tree from one of the parents and substituting it into the other parent by replacing one sub-tree originating from one chosen node. This defines

the crossover of trees to produce offspring. The offspring produced could have a larger complexity, or lower complexity after simplifying the expression. Note that the larger each parent trees are, the more combinations of crossovers there can be. We could define the relationships between offspring that are produced from the same parents to be 'siblings', since they carry similar 'genetic material' in their trees.

With the two core components present, we could employ genetic programming on inflationary potentials. In the next subsections, we demonstrate examples of crossover operations between the function classes defined in (5) and (6).



(a) Tree expression of $c_1 + c_2 * \sin(c_3 * \phi)$ (b) Tree expression of $c_1 + c_2 * \arctan(c_3 * \sin(c_4 * \phi))$

Figure 1: Two tree expressions for two different symbolic expressions

3.2.1 Crossover of Function Classes

Referring to figure 1a, choices of sub-trees to perform crossover include: 1) c, 2) ϕ, 3) $[*, c, \phi]$, 4) $[\sin, *, c, \phi]$, 5) $[*, c, \sin, *, c, \phi]$, 6) $[+, c, *, c, \sin, *, c, \phi]$. These sub-trees can be placed in any of the eight nodes, resulting in a total of 48 siblings. However, some of these may simplify to the same expression, so they may not be distinct. A list of the simplified unique expressions from all possible crossovers can be found in the appendix.

If we move the entire tree of (5) to replace the 'ϕ' node and its sub-tree of (6), we get

$$c_1 + c_2 * \arctan\Big(c_3 * (c_4 + c_5 * \sin(c_6 * \phi))\Big) \longrightarrow c_1 + c_2 * \arctan\Big(c_3 + c_4 * \sin(c_5 * \phi)\Big) \tag{7}$$

A simple redefinition and simplification give a function class

$$c_1 * \arctan(c_2 * \sin(c_3 * \phi)) \tag{8}$$

We will examine this function class more closely in the next section.

3.2.2 Mutation of Function Classes

While the number of possible crossovers between two functions is finite, the number of possible mutations for a node is unlimited. Mutation involves replacing a sub-tree with an arbitrary tree as long as the final expression can be parsed. The simplest kind of mutation would be changing a terminal node to another node, which results in no changes in the overall complexity. The next simplest could be changing a terminal node to two nodes, represented by a unary operator together with its argument. In general, when a non-terminal node gets mutated, the overall complexity could decrease or increase. A tree could always mutate into another tree if the mutation happens at the root level, resulting in virtually infinite possibilities. In this paper, we explore a mutation mechanism inspired by symbolic regression [9].

Symbolic regression (SR) is a type of regression that searches the given space of symbolic expressions to find the models that best fit a given data, in terms of accuracy and simplicity. A function class can be mutated into another by first selecting a random function from that class, and then fitting an SR to it. The best results returned would be functions that are mathematically similar to the selected function. We could extract these functions back to their general function class, and assign probabilities for the original function class to mutate into these function classes accordingly. This mechanism could potentially mutate a function class into another that is mathematically more similar than a purely random mutation into some arbitrary function classes. This happens by remaining the 'genetic properties' in terms of the shape of the functions. The full implementation will be discussed in Sec 5.

4 Breeding of Natural Inflation and Artan Inflation

Natural Inflation [10], [11] and Artan Inflation [12], [13] have been studied extensively, whose inflationary potentials can be expressed and rewritten as

$$V_{natural} = M^4\Big[1 + \cos(\frac{\phi}{f})\Big] \longrightarrow c_1 + c_2 * \cos(c_3 * \phi) \tag{9}$$

$$V_{arctan} = M^4\Big[1 - \frac{2}{\pi}\arctan(\frac{\phi}{\mu})\Big] \longrightarrow c_1 + c_2 * \arctan(c_3 * \phi) \tag{10}$$

The scale M is determined by the CMB normalization and f, μ are unknown scales. Note that they are precisely the expressions of (5) and (6) (sine and cosine could be interchanged under a phase change). As the viability of these two inflationary functions has been discussed in other articles, it is natural to assign some 'good' fitness values to these two function classes such that they have a high probability to reproduce under the genetic programming regime. While this may not be entirely true, it does not prevent the theoretical description of such reproductions between these two function classes.

For the purpose of this paper, we would be looking at a function class defined as

$$\text{ats}_{(c_1,c_2,c_3)}(\phi) \equiv c_1 * \arctan(c_2 * \sin(c_3 * \phi)) \tag{11}$$

which is equation (8). By varying the new coefficients c_1, c_2, c_3, different functions can be formed, and we can write a tuple (c_1, c_2, c_3) to represent such functions. A fitness value can then be assigned to each of these tuples. We would like to find tuples that have fitness zero, as they are possible candidates for inflationary potential.

4.1 A first look at the $\text{ats}_{(c_1,c_2,c_3)}(\phi)$ function class

The function (ArcTan-Sin) $\text{ats}_{(c_1,c_2,c_3)}(\phi)$ is defined in this paper by

$$\text{ats}_{(c_1,c_2,c_3)}(\phi) \equiv (c_1 * \arctan(c_2 * \sin(c_3 * \phi))) \tag{12}$$

where $c_1, c_2, c_3 \in \mathbb{R}$ are three real coefficients. Evidently, it is anti-symmetric and periodic. A relabelling of axes could center either the trough or the peak at $\phi = 0$. A simple redefinition of ϕ by choosing another reference point could transform it into

$$\overline{\text{ats}_{(c_1,c_2,c_3)}(\phi)} \equiv c_1 * [\arctan(c_2) + \arctan(c_2 * \sin(c_3 * \phi \pm \pi/2))], \quad c_1, c_2, c_3 > 0 \tag{13}$$

The $\text{ats}_{(c_1,c_2,c_3)}(\phi)$ and the $\overline{\text{ats}_{(c_1,c_2,c_3)}(\phi)}$ with the exact same coefficients will return the same fitness when passed to the fitness function. This is because the fitness function has a built-in part to translate the functions such that the minimum point is at zero, and translate the whole curve to above $y = 0$.

Next, we would like to investigate each coefficient's role in determining the shape of potential. By inspection, c_1 is a normalization factor used to adjust the magnitude of the core part of the composite function which encloses more oscillatory properties. The steepness of the cliff could be varied with the coefficient c_2 because it quickly amplifies the oscillatory properties of the sin function. If one is to alter $c_2 \in \mathbb{R}_{\geq 0}$, the shape of the function changes drastically at $c_2 \approx O(4)$. The coefficient c_3 is a parameter to rescale the ϕ axis, it tells how many cycles could fit within a reference interval.

4.2 Binary Encoding of c_1, c_2, c_3

As stated in Sec 3.1, we can utilize GA to explore the parameter space of $\text{ats}_{(c_1,c_2,c_3)}(\phi)$ to identify viable potentials. To facilitate this search, we encode each tuple as a binary chain. The range of intervals at which the coefficients are varied and searched is shown in Table 1, and each interval is discretized into 64 equally spaced values using a logarithmic or linear scale. The chain is represented by a repeating unit of $[c_1, c_2, c_3, X]$, and the c_i labels indicate this gene is used for determining the final value of c_i. Each coefficient receives a contribution from its corresponding 6 nodes, each of which can be represented by either 1 or 0. When decoded, the 6 nodes together give an integer ranging from 0 to 63 for that coefficient, which determines its position within the interval varied. X acts as a redundant node and is not used for determining any values for the coefficients. The motivation for such a setup is that having a few redundant nodes accelerate the rate of evolution to the desired regions. While the justification is not rigorous, it produces a larger number of viable potentials in the trial runs. Other encoding methods exist, as long as the encoding is able to identify a large number of solutions. The schematic allele diagram used by the GAs is shown in Figure 2.

coefficient	range	no. of partitions	scale
c_1	$[10^{-13}, 10^{-9}]$	64	logarithmic
c_2	$[0, 200]$	64	linear
c_3	$[0, 5]$	64	linear

Table 1: Boundaries of the parameter space the genetic algorithms searched on

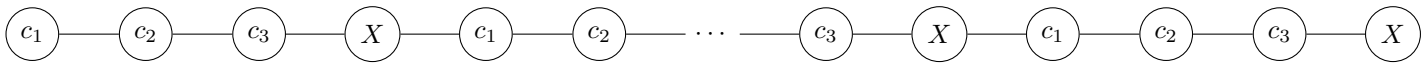


Figure 2: The binary allele string diagram of the periodic placements of nodes contributing to the parameters.

4.3 Genetic Algorithm Search

In each new run, a random population ($N_{pop} = 100$) is initialized, with each individual consisting of a 24-bit long binary list. The algorithm arranges the functions in descending order of fitness value and selects them for breeding based on a probability given by

$$P_k = \frac{2}{(1+\alpha)N_{pop}}\left(1 + \frac{N_{pop} - k}{N_{pop} - 1}(\alpha - 1)\right) \tag{14}$$

where k represents the rank of the function, and α is chosen to be 3.5 This configuration is thought to accelerate the chance of obtaining functions with higher fitness values by letting the fitter functions have a higher chance of breeding by mimicking natural selection. The probability P_1 for the function that ranked the first is α times the probability of the least fit function. The breeding process used in this paper is a one-point crossover, where two parent strings are divided into two substrings at the same node, and the offspring are produced by merging two substrings from different parents together. Mutation occurs within each node of the offspring with a chosen probability of 10%, and when it happens, the value of the node swap between 1 and 0. These processes are repeated 100 generations in each run, resulting in the exploration of 10000 functions in each run.

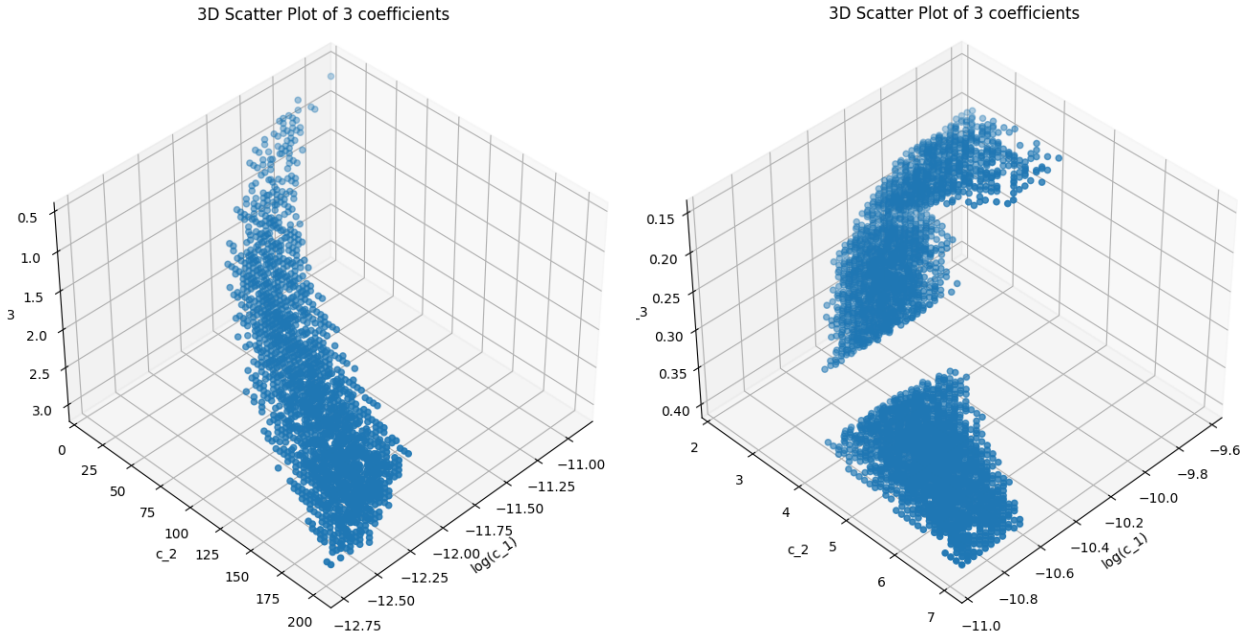
4.4 Results

In Sec 4.3, the configuration was run 13 times, resulting in the identification of 1768 distinct tuples out of 130,000 states that were visited. With the 3 axes chopped into 64 partitions each, there are a total of $64^3 = 262144$ possible states. While 130,000 states have been visited, this does not mean they are distinct. However, with only 1768 distinct viable states, it is enough to give us interesting statistics. Figure 3a shows a 3d scatter plot of the tuples within the search region, which appears as an arm with two ends. One end extends to larger c_2, c_3 values, while the other end truncates at smaller c_2, c_3 values. Further refinement of the structures at the smaller end was carried out with another 13 runs of GA in the region defined in table 2, which resulted in the identification of 3565 distinct tuples. Figure 3b shows another 3d scatter plot of the tuples found in this smaller region, where a distinct gap opens up between two clusters of points.

To examine the nature of this gap, solutions with $-10.6 < \log c_1 < -10.4$ is plotted in a plane with c_3 against c_2 in figure 4a. By choosing $\log c_1 = -10.5$, examine five different combinations of c_2, c_3 labeled $f_i, i = 1, 2, 3, 4, 5$ sampled from a red dashed line connecting the two clusters. The first two functions (f_1, f_2) belongs to the cluster with relatively smaller c_2, c_3 (cluster 1), f_3 is located within the gap, and (f_4, f_5) belong to the cluster with relatively larger c_2, c_3 (cluster 2). The choices of (c_2, c_3) are given and the five functions are plotted in figure 4b, with the respective rolling regions colored in black. It can be seen that the rolls from cluster 1 are relatively larger than those from cluster 2. In fact, most curves in cluster 2 have their rolling region located in the metastable plateau only, corresponding to small field inflation. On the other hand, curves in cluster 1 roll down a large portion of the curve, and correspond to large field inflation. Those within the gap are non-viable since the roll and the number of e-folds are too small to resolve the horizon problem. A crude classification of the two classes can be decided by $c_2 \approx 4$, where $c_2 \gtrapprox 4$ is small field inflation and $c_2 \lessapprox 4$ is large field inflation. One possible interpretation of these results is that the shape of the $\text{ats}_{(c_1,c_2,c_3)}(\phi)$ changes drastically at $c_2 \approx O(4)$, which is approximately the transition between the sin regime ($c_2 \lessapprox 4$) and the step function regime ($c_2 \gtrapprox 4$). In between the transitions, the slow-roll region is squeezed into a minimal region on the slope which has a large first derivative, thus violating the slow-roll approximations easily. The constraint imposed by the number of e-folds leads to the existence of this gap by cutting it open.

As of the writing of this paper, it is uncertain whether the arm has an end on the part of the larger cluster. By varying the range of the GAs to operate on, it was found that c_2 could go up to $O(10^4)$ and c_3 going up to $O(10^2)$, and yet still be viable. It is not known whether a smaller value of $\log c_1$ would bring a natural end to this part of the arm. A GIF file of the

3d view on the parameter spaces from different angles could be found here [14]. The code used and results are included in the GitHub repository as well.



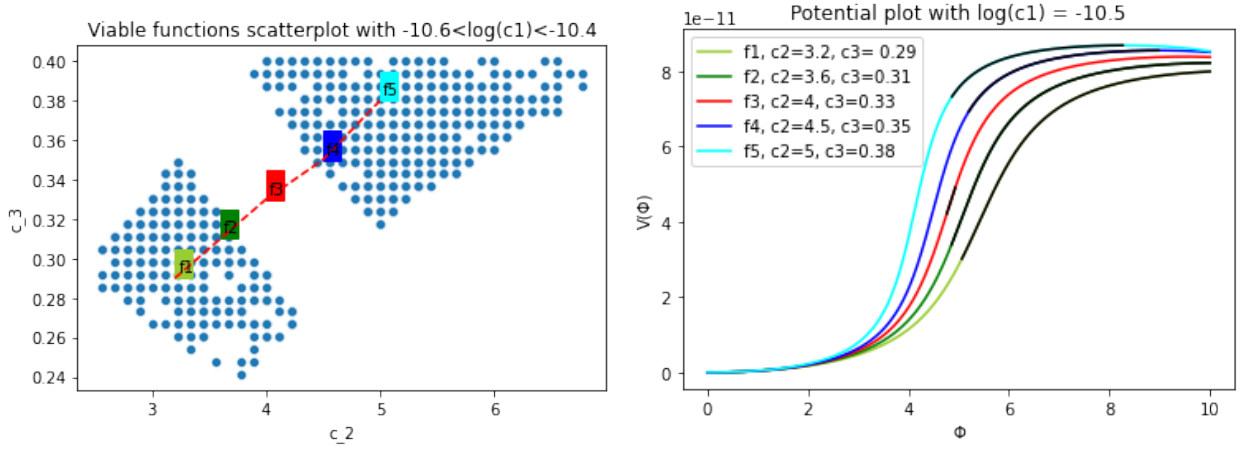
(a) Scatter plot within region labeled by table 1

(b) Scatter plot within region labeled by table 2

Figure 3: 3d scatter plots of the tuples from the two different search spaces.

coefficient	range	no. of partitions	scale
c_1	$[10^{-11}, 10^{-9}]$	64	logarithmic
c_2	$[0, 7]$	64	linear
c_3	$[0, 0.4]$	64	linear

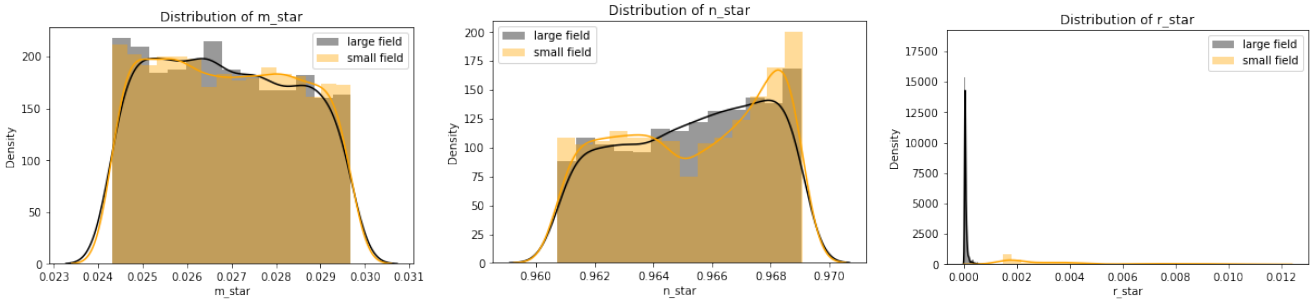
Table 2: Second set of boundaries of the parameter space the genetic algorithms searched on



(a) Scatter plot in the plane of c_2, c_3 with $-10.6 < \log(c_1) < -10.4$

(b) Plot of the five sample functions $(f_1, f_2, f_3, f_4, f_5)$, with the rolling region in each function colored in black.

Figure 4: Graphic display of samples from the results.



(a) The energy scale m_* (b) The spectral index $n_{s,*}$ (c) The tensor-to-scalar ratio r_*

Figure 5: Density plot of three cosmological observables between the large field (coloured in grey) and small field inflation (coloured in orange) of the $\text{ats}_{(c_1,c_2,c_3)}(\phi)$ function.

After identifying the two clusters, we could compare their cosmological predictions as different forms of potential can have varying cosmological attributes. Specially, we analyzed the spectral index n_s, energy scale m, and tensor-to-scalar ratio $r.$, while disregarding the distribution of the number of e-folds, as long as it is larger than the reference value. Figure 5 displays the density plots of the three observables. The energy scale m_* is distributed quite uniformly over the range for both large and small-scale inflation. On the other hand, both large and small-scale inflation favor a spectral index with $n_{s,*} \approx 0.9685$. However, the distribution of the tensor-to-scalar ratio r_* differs between the two types of inflation. Large-scale inflation has this value centered at around 0.002, while that of the small-scale is centered at around 0.00005. This may be due to the fact that the two types of inflation roll at significantly different regions, as argued by their partition by a gap. Comparing these results with Ref [3] where the distribution of r_* has been studied for the class of polynomial potentials and is centered at around $r_* = 0.0002$, the two new classes studied are about 10 folds larger or smaller than that.

4.5 Function Relatives

So far we have only considered the $\text{ats}_{(c_1,c_2,c_3)}(\phi)$ function class. It is natural to ask whether there are infinitely many classes of viable functions, if so, what are their respective parameter spaces? We could start by looking at its 'genetic-relatives' when expressed in a tree diagram.

$$c_1 * \arctan(c_2 * \sin(c_3 * x)) \longleftrightarrow [\ *\ ,\ c_1\ ,\ \arctan\ ,\ *\ ,\ c_2\ ,\ \sin\ ,\ *\ ,\ c_3\ ,\ \mathrm{x}\ ] \tag{15}$$

$$c_1 * \arctan(c_2 * \cos(c_3 * x)) \longleftrightarrow [\ *\ ,\ c_1\ ,\ \arctan\ ,\ *\ ,\ c_2\ ,\ \cos\ ,\ *\ ,\ c_3\ ,\ \mathrm{x}\ ] \tag{16}$$

$$c_1 * \arctan(c_2 * \sin(c_3 * x)) + c_4 \longleftrightarrow [\ +\ ,\ *\ ,\ c_1\ ,\ \arctan\ ,\ *\ ,\ c_2\ ,\ \sin\ ,\ *\ ,\ c_3\ ,\ \mathrm{x}\ ,\ c_4] \tag{17}$$

$$c_1 * \arctan(c_2 * \sin(c_3 * x + c_4)) \longleftrightarrow [\ *\ ,\ c_1\ ,\ \arctan\ ,\ *\ ,\ c_2\ ,\ \sin\ ,\ +\ ,\ *\ ,\ c_3\ ,\ \mathrm{x}\ ,\ c_4] \tag{18}$$

$$c_1 * \arctan(c_2 * \sin(c_3 * x - c_4)) \longleftrightarrow [\ *\ ,\ c_1\ ,\ \arctan\ ,\ *\ ,\ c_2\ ,\ \sin\ ,\ \text{-}\ ,\ *\ ,\ c_3\ ,\ \mathrm{x}\ ,\ c_4] \tag{19}$$

$$c_1 * \arctan(c_2 * \sin(c_3 * x) + c_4) \longleftrightarrow [\ *\ ,\ c_1\ ,\ \arctan\ ,\ +\ ,\ *\ ,\ c_2\ ,\ \sin\ ,\ *\ ,\ c_3\ ,\ \mathrm{x}\ ,\ c_4] \tag{20}$$

Consider the case where c_1, c_2, c_3 is taken from the viable set, so (15) is viable. (16) differs from (15) with the node sin $\rightarrow$ cos, and is a viable function by relabelling the ϕ-axis. (15) and (17) differ by the addition of a parameter c_4, and increase the complexity of the function. Since the fitness function translates the curve to above zero and center at the origin, c_4 could be any real number and (17) is viable. (15) differs from (18) in a similar way as that of (15) and (16), and again c_4 could be any real number by relabelling the ϕ axis so (18) is viable. (19) is viable by a parity transform on (18). Consider the condition $|c_4| \ll |c_3 * x|$, (20) essentially simplifies to (15), so its viable region is a bunch of viable regions of (15) sticking together around $c_4 \approx 0$. These are just a few examples of functions that have similar tree structures with (15) and share similar viable topologies within the parameter space. We have shown that for some functions, the topology of the viable regions between functions with minor differences in the tree representation could be similar.

5 Symbolic Regression as Mutation

In Sec 3.2.2, we discussed Symbolic Regression (SR) algorithms, which learn analytic expressions to fit data symbolically. Specifically, we would examine the method of Exhaustive Symbolic Regression (ESR) [9]. First, we would like to define all possible trees of expressions. By specifying the operators (binary, unary, nullary) and the maximum complexity of a tree, ESR generates all unique expressions up to the maximum complexity using the basis functions defined by the operators. The

ESR then fits all these unique equations with the data by utilizing the minimum description length principle, ranking all the function classes with fitted parameters by measuring both how well the function fits the data and penalizing more complex functional forms. This is encapsulated in a single value called the 'code length' L(D). The smaller it is, the better it fulfills the minimum length principle.

By selecting the basis functions shown in table 3, set the maximum complexity of any tree to be 6. Around 4000 function classes are fitted against the function $7.74 * 10^{-11} * \arctan(2.89 * \sin * (0.28 * \phi))$ which belongs to cluster 1. Table 4 displays the top 5 results returned by the algorithm. We proceed to examine the fitness of the top-ranked function, which is -1.0745 and is not viable. However, if we use the magnitude of the coefficients as a guiding principle for a GA search: by increasing the complexity of this function class to become $c_1 * |\arctan(c_2 * \phi)|^{c_3}$ by giving it a prefactor c_1 to control the magnitude of the potential, and perform a GA run over the interval given by table 5. A total of 599 viable potentials from this new function class have been identified. This demonstrates that using symbolic regression as a means of mutation of function classes could be used to identify viable potentials with new functional forms.

As of the writing of this paper, the remaining 4 function classes returned by the ESR have not been examined, and the full properties of the new function class have not been examined closely. Yet, it provides proof that such an ESR mutation mechanism could potentially uncover more inflationary potentials if run with higher complexity or with larger basis functions set. The remaining functions might be viable potentials if given a normalization constant. As the algorithm was run on the author's laptop, only complexity up to 6 could be performed. It has been argued direct viable solutions could be returned if the algorithm is run on complexity 8, or at an even higher complexity.

Basis functions	
Type	Functions
0	'x', 'a'
1	'inv', 'sin', 'arctan'
2	'+', '-', '*', '/', 'pow'

Table 3: Basis functions used

Rank	Function Class	L(D)	a_0	a_1
1	$\lvert\arctan(a_0 * \phi)\rvert^{a_1}$	48945.66	$7.85 * 10^{-1}$	7.75
2	$\lvert\arctan(a_0 + \phi)\rvert^{a_1}$	63532.41	$4.28 * 10^{-7}$	7.38
3	$\lvert a_0 + \arctan(\phi)\rvert^{a_1}$	63532.65	$4.34 * 10^{-9}$	7.38
4	$\lvert\arctan(\frac{a_0}{\phi})\rvert^{a_1}$	108657.28	$4.46 * 10^{-1}$	$-8.39 * 10^{-1}$
5	$a_0 * \phi * \arctan(\phi)$	111346.37	1.08	0.00

Table 4: Top 5 ESR results on the selected function

coefficient	range	no. of partitions	scale
c_1	$[10^{-10}, 10^{-8}]$	64	logarithmic
c_2	$[0.2, 1]$	64	linear
c_3	$[6, 9]$	64	linear

Table 5: Boundaries of the coefficient for the $c_1 * |\arctan(c_2 * \phi)|^{c_3}$ function class

These results suggest that one could synthesize inflationary potentials by generating a series of points clouds around a known viable potential, and then examine the results returned by a symbolic regression on these data points. Future work could focus on the point clouds generation mechanism and testing them with higher complexity with the use of a computer cluster.

6 Conclusion

This paper presents a novel genetic programming approach to identify inflationary potentials through the breeding of natural inflation and arctan inflation. The criteria for determining the viability of a function are based on its ability to produce consistent cosmological predictions, including the number of e-folds, the spectral index n_s, the tensor-to-scalar ratio r_* and the energy scale of inflation m_*. Evolutionary algorithms are reviewed and applied to generate the $\text{ats}_{(c_1,c_2,c_3)}(\phi)$ function class. The topology of the viable regions within its parameter space is explored, two clusters of solutions are found and compared with each other by investigating some of their samples. Finally, a mutation mechanism using exhaustive symbolic regression is introduced to discover a new function class $(c_1 * |\arctan(c_2 * \phi)|^{c_3})$ with viable potentials, demonstrating the potential for genetic programming to identify new inflationary potentials. Future work can focus on further exploring the properties of the new function class and scaling up the genetic programming approach to a larger population with more computing resources. Other mutation mechanisms could be considered. For example, instead of fitting the chosen function

to ESR, we could replace part of the function with a series of 'point clouds', which could serve similar purposes and guide the algorithm towards new function classes. This could be explored in an upcoming paper.

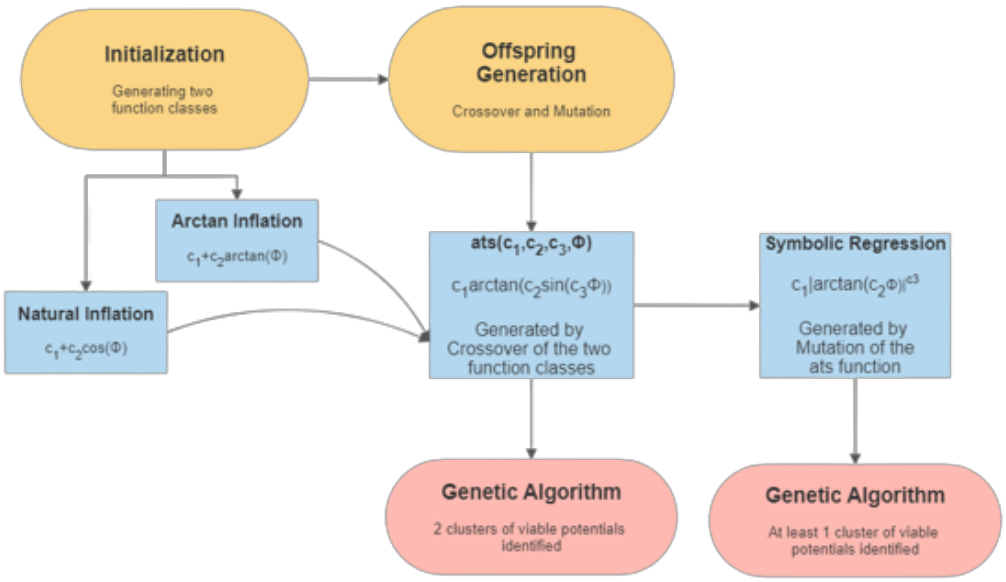


Figure 6: Summary of the Genetic Programming employed in identifying new inflationary potentials.

References

[1] D. Baumann and L. McAllister, *Inflation and string theory*, May 2015. DOI: 10.1017/CBO9781316105733.

[2] D. Baumann, *Tasi lectures on inflation*, 2012. DOI: https://doi.org/10.48550/arXiv.0907.5424.

[3] S. Abel, A. Constantin, T. R. Harvey, and A. Lukas, *Cosmic inflation and genetic algorithms*, 2022. DOI: 10.48550/ARXIV.2208.13804.

[4] A. Kamerkar, S. Nesseris, and L. Pinol, *Machine learning cosmic inflation*, 2022. DOI: 10.48550/ARXIV.2211.14142. [Online]. Available: https://arxiv.org/abs/2211.14142.

[5] J. Martin, C. Ringeval, and V. Vennin, "Encyclopaedia inflationaris," 2013. DOI: 10.48550/arXiv.1303.3787.

[6] Y. Akrami, F. Arroja, M. Ashdown, *et al.*, "Planck 2018 results. x. constraints on inflation," *Astronomy & Astrophysics*, vol. 641, A10, Sep. 2020. DOI: 10.1051/0004-6361/201833887.

[7] E. F. Bunn, A. R. Liddle, and M. White, "Four-year cobe normalization of inflationary cosmologies," *Phys. Rev. D*, vol. 54, R5917–R5921, 10 Nov. 1996. DOI: 10.1103/PhysRevD.54.R5917.

[8] S. Weinberg, *Cosmology*. OUP Oxford, 2008, ISBN: 9780191523601.

[9] D. J. Bartlett, H. Desmond, and P. G. Ferreira, "Exhaustive Symbolic Regression," Nov. 2022. [Online]. Available: https://arxiv.org/abs/2211.11461.

[10] F. C. Adams, J. R. Bond, K. Freese, J. A. Frieman, and A. V. Olinto, "Natural inflation: Particle physics models, power-law spectra for large-scale structure, and constraints from the cosmic background explorer," *Phys. Rev. D*, vol. 47, pp. 426–455, 2 Jan. 1993. DOI: 10.1103/PhysRevD.47.426.

[11] K. Freese, J. A. Frieman, and A. V. Olinto, "Natural inflation with pseudo nambu-goldstone bosons," *Phys. Rev. Lett.*, vol. 65, pp. 3233–3236, 26 Dec. 1990. DOI: 10.1103/PhysRevLett.65.3233.

[12] L. Wang, V. Mukhanov, and P. J. Steinhardt, "On the problem of predicting inflationary perturbations," *Physics Letters B*, vol. 414, no. 1-2, pp. 18–27, Nov. 1997. DOI: 10.1016/s0370-2693(97)01166-0.

[13] S. M. Leach, A. R. Liddle, J. Martin, and D. J. Schwarz, "Cosmological parameter estimation and the inflationary cosmology," *Physical Review D*, vol. 66, no. 2, Jul. 2002. DOI: 10.1103/physrevd.66.023515.

[14] "GitHub repository," [Online]. Available: https://github.com/Coding-appreciator/ESR.

[15] D. H. Lyth and A. Riotto, "Particle physics models of inflation and the cosmological density perturbation," *Physics Reports*, vol. 314, no. 1-2, pp. 1–146, Jun. 1999. DOI: 10.1016/s0370-1573(98)00128-8.

Appendix A Fitness Function

The fitness function is defined as

$$f = f_N + f_{n_s} + f_r + f_m \tag{21}$$

where each term penalizes the function if any of the predictions are way off the limits observed. f_N measures the fitness of the total number of e-folds, f_{n_s} the fitness of the spectral index, f_r the fitness of the tensor-to-scalar ratio, and f_m the fitness of the energy scale of inflation.

The total number of e-folds should be larger than the reference value of e-folds which separates the end of inflation ϕ_e from the reference value ϕ_* when the cosmological scales probed by the Planck satellite leave the horizon. This value can be approximated by [15]

$$N_* \approx 58 + \frac{1}{6}\ln V(\phi_e) \tag{22}$$

The value of the field ϕ_* at the pivot scale is obtained by the integral equation

$$N_* = \int_{\phi_e}^{\phi_*} \frac{V}{V'} d\phi \tag{23}$$

The total number of e-folds N_{total}, the spectral index $n_{s,*}$, tensor-to-scalar ratio r_*, and the energy scale m_* at the pivot scale are given by

$$N_{total} = \Big| \int_{\phi_e}^{\phi_i} \frac{V}{V'} d\phi \Big|, \quad n_{s,*} = 1 - 6\epsilon_* + 2\eta_*, \quad r_* = 16\epsilon_*, \quad m_* = \Big(\frac{V_*}{\epsilon_*}\Big)^{1/4} \tag{24}$$

With $\theta(x)$ being the Heaviside function, f_N is chosen to be

$$f_N = -\frac{1}{10}(N_* - N_{total}) \cdot \theta(N_* - N_{total}) \tag{25}$$

The contribution f_{n_s} is chosen to be

$$f_{n_s} = \hat{f_{n_s}} \cdot \theta(f_{term} - \hat{f_{n_s}}) \tag{26}$$

where

$$\hat{f_{n_s}} = -\log_{10}\Big(1 + \frac{|n_{s,*} - n_s|}{\Delta n_s}\Big) \tag{27}$$

Note that $n_s \pm \Delta n_s = 0.9649 \pm 0.0042$, f_{term} is chosen to be -0.3 such that $\hat{f_{n_s}}$ attracts a vanishing contribution to the total fitness if the predicted value of n_s is within 5% of the current measurements.
Similarly, the tensor-to-scalar ratio r_* is fixed by $r_{max} = 0.061$.

$$f_r = -\log_{10}\Big(\frac{r_*}{r_{max}}\Big) \cdot \theta(r_* - r_{max}) \tag{28}$$

And finally the fitness contribution from the energy scale

$$f_m = \hat{f_m} \cdot \theta(f_{term} - \hat{f_m}) \tag{29}$$

where

$$\hat{f_m} = -\log_{10}\Big(1 + \frac{|m_* - m|}{\Delta m}\Big) \tag{30}$$

f_{term} is still chosen as -0.3 and $m \pm \Delta m = 0.027 \pm 0.0027$.

Appendix B Functions

This table shows all the unique crossovers between natural inflation and arctan inflation.

c, $c_1 + c_2\arctan(c_3\phi)$, $c_1 + c_2\arctan(c_3 + \phi)$, ϕ, $\phi + c_1\arctan(c_2\phi)$, $c_1 + \phi$, $c_1 + c_2\phi$, $c_1 + c_2\arctan(\phi)$,$c_1 + c_2\arctan(\phi^2)$, $c_1 + c_2\arctan(c_3\phi)$, $c_1\phi$, $c_1\phi + c_2\arctan(c_3\phi)$, $c_1 + c_2\phi$, $c_1 + c_2\phi\arctan(c_3\phi)$, $c_1 + c_2\arctan(c_3\phi)$, $c_1 + c_2\arctan(c_3\phi^2)$, $\arctan(c_1\phi)$, $\arctan(c_1\phi) + c_2\arctan(c_3\phi)$, $c_1 + \arctan(c_2\phi)$, $c_1 + \arctan(c_2\phi)\arctan(c_3\phi)$, $c_1 + c_2\arctan(c_3\phi)$, $c_1 + c_2\arctan(\arctan(c_3\phi))$, $c_1 + c_2\arctan(c_3\arctan(c_4\phi))$, $c_1\arctan(c_2\phi)$, $c_1\arctan(c_2\phi) + c_3\arctan(c_4\phi)$, $c_1 + c_2\arctan(c_3\phi)\arctan(c_4\phi)$, $c_1 + c_2\arctan(c_3\arctan(c_4\phi))$, $c_1 + c_2\arctan(c_3\phi\arctan(c_4\phi))$, $c_1 + c_2\arctan(c_3\arctan(c_4\phi))$, $c_1 + c_2\arctan(c_3\phi)$, $c_1 + c_2\arctan(c_3\phi) + c_4\arctan(c_5\phi)$, $c_1 + \big(c_2 + c_3\arctan(c_4\phi)\big)\arctan(c_5\phi)$, $c_1 + c_2\arctan(c_3 + c_4\arctan(c_5\phi))$, $c_1 + c_2\arctan\big([c_3 + c_4\arctan(c_5\phi)]\phi\big)$, $c_1 + c_2\arctan(c_3 + c_4\arctan(c_5\phi))$