

https://vk.com/true_mathbot

```
optimize x + y^2 - 1000.5  
for x in [-10; 100],  
y in (10.3e-100, 123)  
| minorant 10
```

[https://github.com/donRumata03/ITM0.STARS_texts/blob/master/
FinalResults/Short_guide-1_1.pdf](https://github.com/donRumata03/ITM0.STARS_texts/blob/master/FinalResults/Short_guide-1_1.pdf)

<https://github.com/donRumata03/MathBotBackend>
<https://github.com/donRumata03/pythonich>
<https://github.com/donRumata03/PowerfulGA>
<https://github.com/donRumata03/ExpressionParsing>
<https://github.com/donRumata03/MathBotFrontend>

```

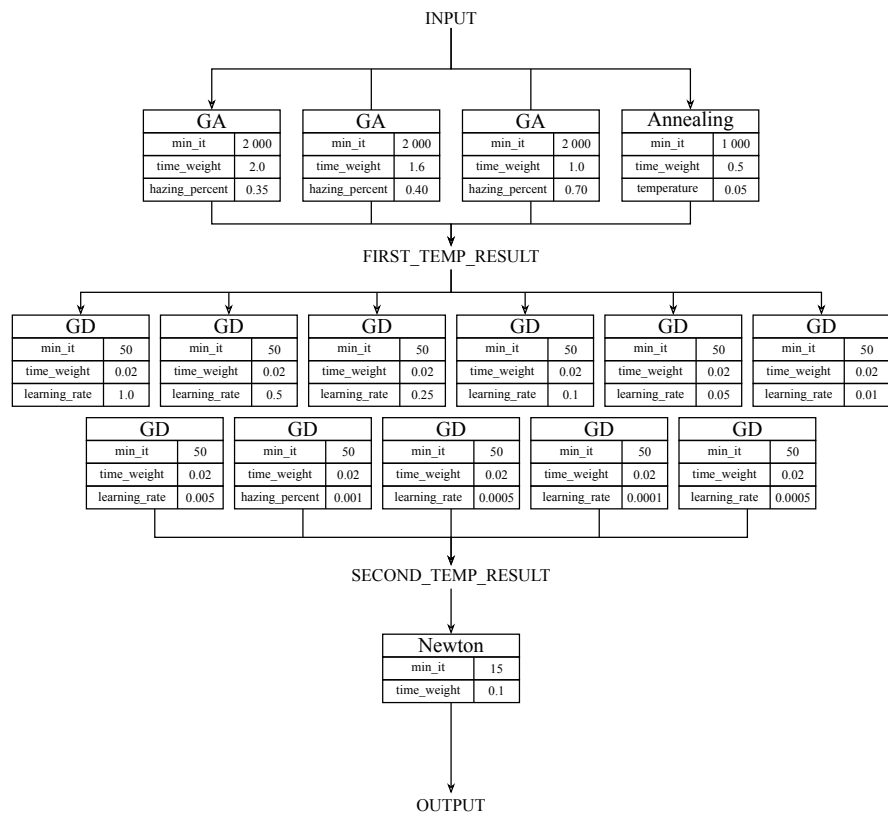
case operation_char::addition: {
    // Check if the exponent is an integer constant
    if (right.type == type::number) {
        auto exponent = static_cast<long long>(std::round((double)right.compute()));
        if (double(exponent) == right.compute()) {
            return new expression_tree(
                operation_char::multiplication,
                left->generate_derivative_tree(diff_variable_name),
                right->new_expression_tree(
                    operation_char::multiplication, double(exponent),
                    right->new_expression_tree(
                        operation_char::addition,
                        left->new_expression_tree(
                            operation_char::subtraction,
                            left->new_expression_tree(
                                double(exponent)),
                                number(1.))));
            }
        }
    }
    auto log_tree = std::make_unique<expression_tree>(
        operation_char::multiplication, left->new_expression_tree(right),
        right->new_expression_tree(
            operation_char::natural_logarithm, left->new_expression_tree(
                operation_char::multiplication, left->new_expression_tree(
                    this), log_tree->generate_derivative_tree(diff_variable_name));
    }
}

```

$$10 + x + 11 * \exp(3) \longrightarrow (10 + 11 * \exp(3)) + x \longrightarrow (\approx 230.94) + x$$

$$x + 0 = x x^1 = x$$



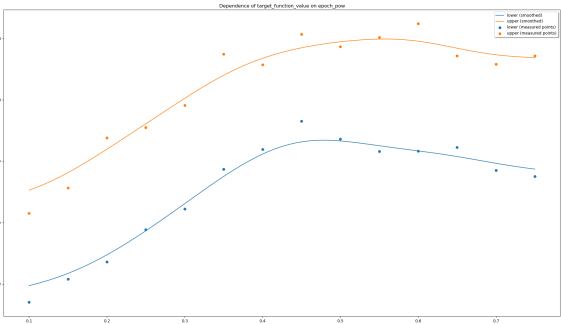


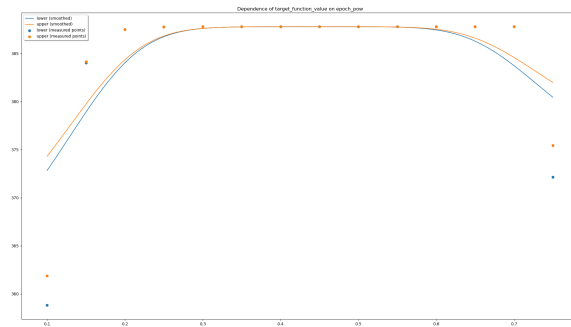
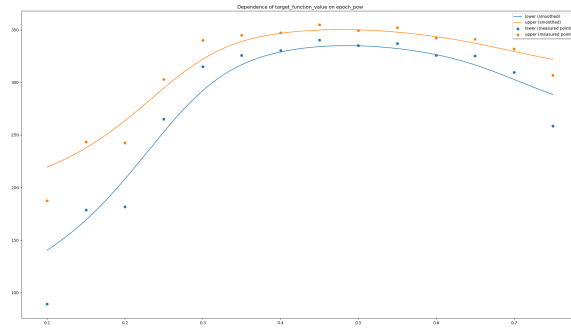
$$\frac{E \times population\ size}{E} = population\ size$$

$$\begin{cases} E = computational\ resource^{epoch\ pow} \\ population\ size = \frac{computational\ resource}{E} = computational\ resource^{1-epoch\ pow} \end{cases}$$

$$M(x_1,\ldots,x_n)=\varphi^{-1}\left(\frac{1}{n}\sum_{k=1}^n\varphi(x_k)\right)$$

$$\varphi(x)=x^{-1.5}\varphi(x)=x^{2.5}$$





$epochpow \approx 0.45$

https://en.wikipedia.org/wiki/Test_functions_for_optimization

```
solve (eq_1_left - eq_1_right)^2
+ ... + (eq_n_left - eq_n_right)^2
for x in [...; ...],
y in (...; ...),
```

...
| minorant ...