

Iz tabele sem izbral 3 sintetične funkcije

Koza-3 [33] 1: $x^6 - 2x^4 + x^2$ U[-1, 1, 20] None Koza

Nguyen-5 [45] 1: $\sin(x^2)\cos(x) - 1$ U[-1, 1, 20] None Koza

Nguyen-10 [45] 2: $2\sin(x)\cos(y)$ U[-1, 1, 100] None Koza

Prva implementacija je imale enake

```
//Define base function node types
List<Class<? extends Node>> baseFunctionNodeTypes = Arrays.asList(
    AddNode.class,
    DivNode.class,
    MulNode.class,
    SubNode.class,
    SinNode.class,
    CosNode.class
);

//Define base terminal node types
List<Class<? extends Node>> baseTerminalNodeTypes = Arrays.asList(
    ConstNode.class,
    VarNode.class
);
```

--- KOZA3 ---

Press E to pause the algorithm:

Fitness: 2.746680912396917E-4

Expression:

$$(((\sin((-3.8655441465404428 + x) * \sin(x))) + \cos(\sin((-4.034894601799386 * 6.570779376355219) - (x * x)))) * ((\sin(-2.662570576038819) + \cos(x)) - \sin((\sin(x) - 5.099190916275722)))) * \sin(\sin(x)))$$

--- NGUYEN5 ---

Fitness: 2.746680912396917E-4

Expression:

$$(((\sin((-3.8655441465404428 + x) * \sin(x))) + \cos(\sin((-4.034894601799386 * 6.570779376355219) - (x * x)))) * ((\sin(-2.662570576038819) + \cos(x)) - \sin((\sin(x) - 5.099190916275722)))) * \sin(\sin(x)))$$

--- NGUYEN10 ---

Fitness: 0.0

Expression:

$$((\cos(y) * \sin(x)) + (\cos(y) * \sin(x)))$$

Ločen nastavljeni baseFunctionNodeTypes in baseTerminalNodeTypes za vsak problem

```
List<Class<? extends Node>> FunctionNodeTypes_K0ZA3 = Arrays.asList(  
    AddNode.class,  
    SubNode.class,  
    MulNode.class,  
    SafePowNode.class  
);  
  
List<Class<? extends Node>> TerminalNodeTypes = Arrays.asList(  
    VarNode.class  
);  
  
SymbolicRegressionProblem problem_K0ZA3 =  
    new SymbolicRegressionProblem(  
        FunctionNodeTypes_K0ZA3,  
        baseFunctionNodeTypes,  
        TerminalNodeTypes,  
        baseTerminalNodeTypes,  
        trainingData_K0ZA3  
    );
```

Fitness: 0.007379983600010515

Expression:

$$\begin{aligned} &((((((x \ x) \ (x \ x)) \ ((x \ x) - (x \ x))) * ((x \ (x * x)) * ((x - x) - (x \ x)))) ((x - x) - x) + ((x - x) + (x * x))) * \\ &(((x \ x) - (x + x)) + ((x * x) + (x - x)))) (((((x * x) \ (x \ x)) + ((x \ x) * (x - x))) * (((x - x) * (x \ x)) - ((x + x) \\ &* x))) - x)) - ((x + x) (((x - ((x - x) + x)) - (x - ((x - x) \ (x - x)))) * ((x \ ((x \ x) + x)) * (((x - x) * (x \ x)) - ((x \\ &x) \ (x \ x))))))) \end{aligned}$$

```

List<Class<? extends Node>> FunctionNodeTypes_NGUYEN5 = Arrays.asList(
    AddNode.class,
    SubNode.class,
    MulNode.class,
    SafePowNode.class,
    SinNode.class,
    CosNode.class
);

List<Class<? extends Node>> baseTerminalNodeTypes = Arrays.asList(
    ConstNode.class,
    VarNode.class
);

SymbolicRegressionProblem problem_NGUYEN5 =
    new SymbolicRegressionProblem(
        FunctionNodeTypes_NGUYEN5,
        baseFunctionNodeTypes,
        baseTerminalNodeTypes,
        trainingData_NGUYEN5
    );

```

Fitness: 1.8398243896262398E-5

Expression:

$\sin((3.984098545541908 + (x + \cos(\sin((\cos(x) + (x + 7.573450392168077)))))))$

```

List<Class<? extends Node>> FunctionNodeTypes_NGUYEN10 = Arrays.asList(
    MulNode.class,
    SafePowNode.class,
    SinNode.class,
    CosNode.class
);

SymbolicRegressionProblem problem_NGUYEN10 =
    new SymbolicRegressionProblem(
        FunctionNodeTypes_NGUYEN10,
        baseFunctionNodeTypes,
        baseTerminalNodeTypes,
        trainingData_NGUYEN10
    );

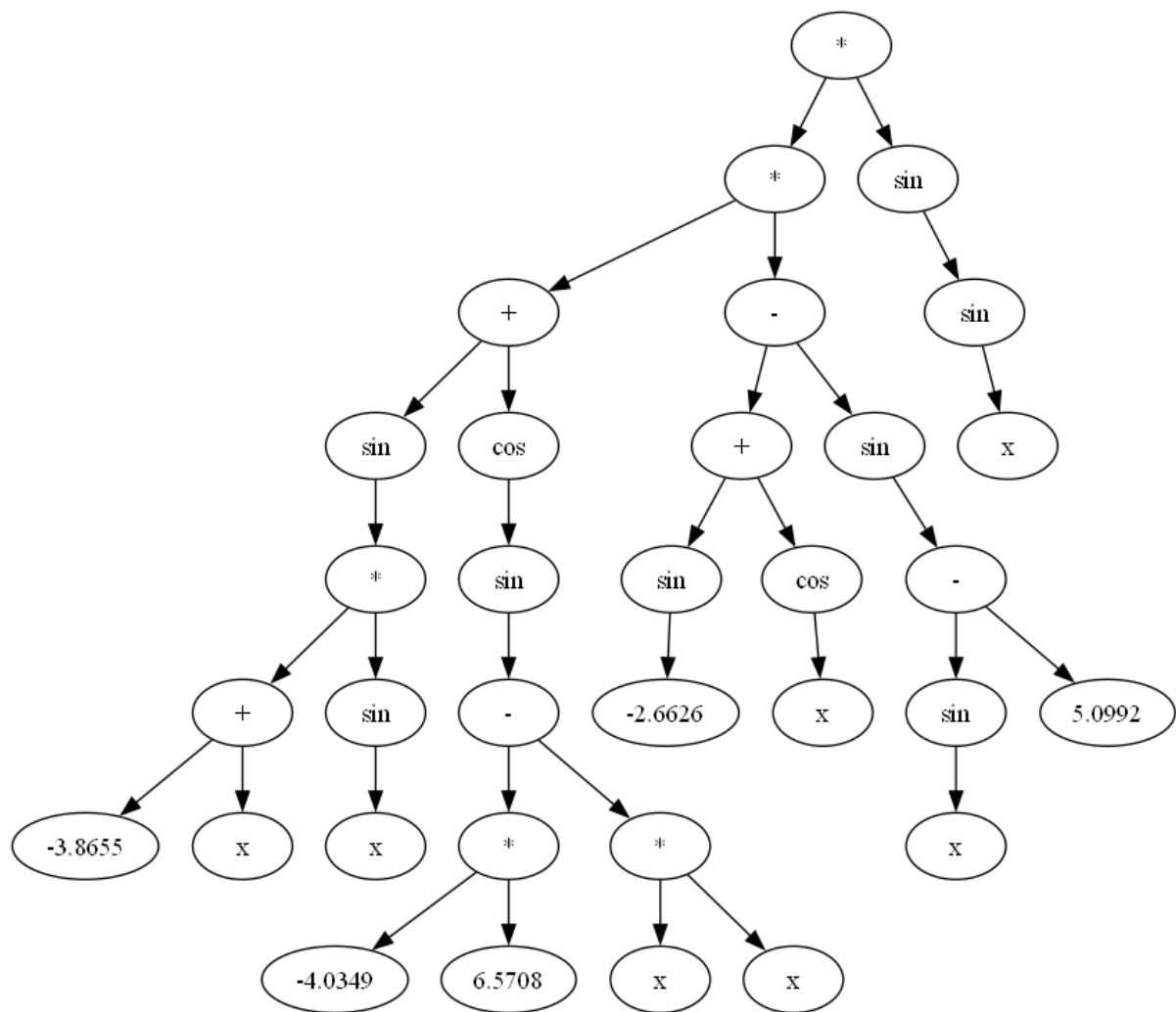
```

Fitness: 1.8245712984995445E-9

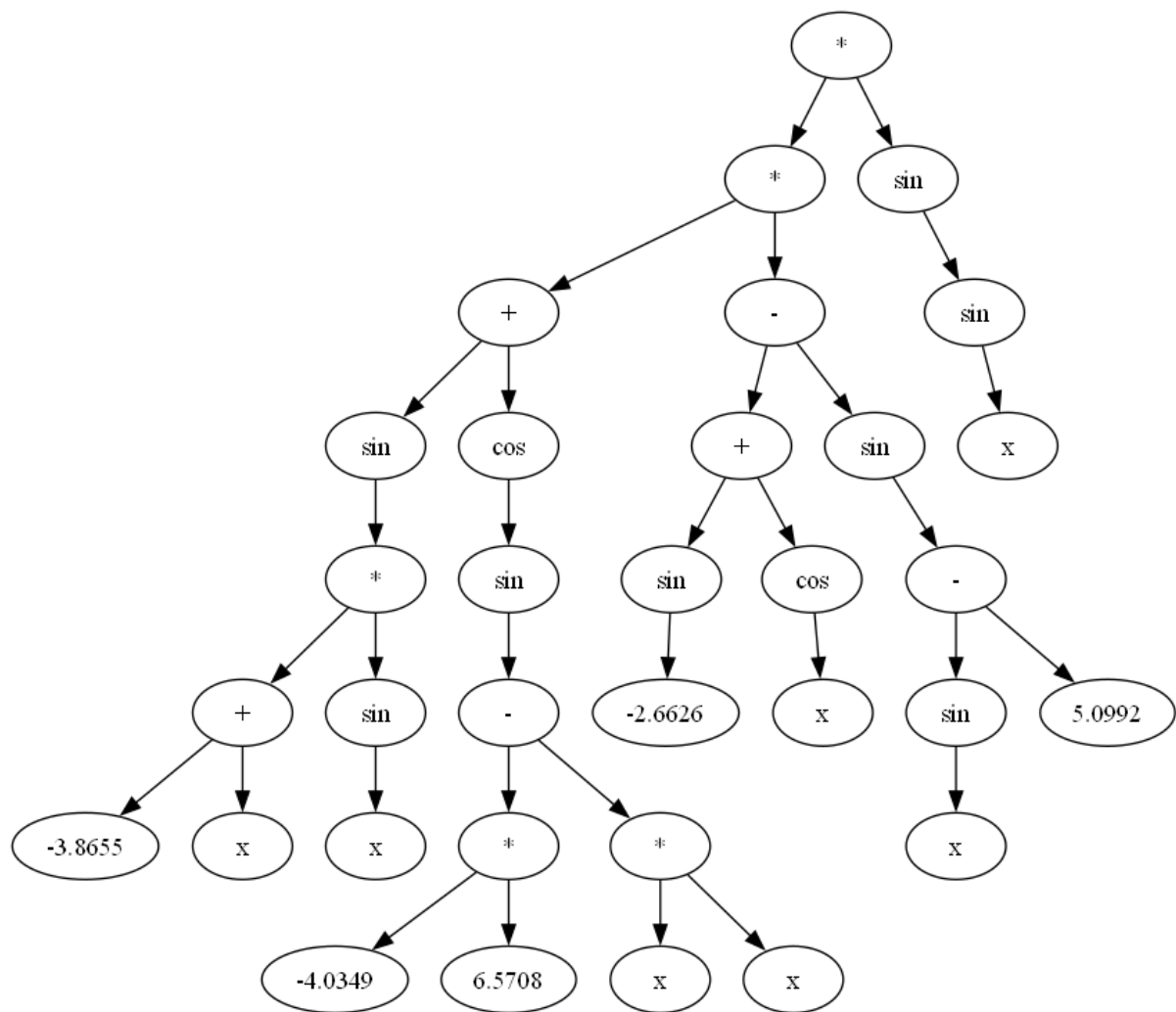
Expression:

$((\cos(\sin(-6.059195467229386)) * ((-6.059195467229386 \cos(x)) * \sin(x))) * ((\cos(y) * \sin(((x \cos(5.28711419907876)) \cos(y)))) * 2.436773029985769))$

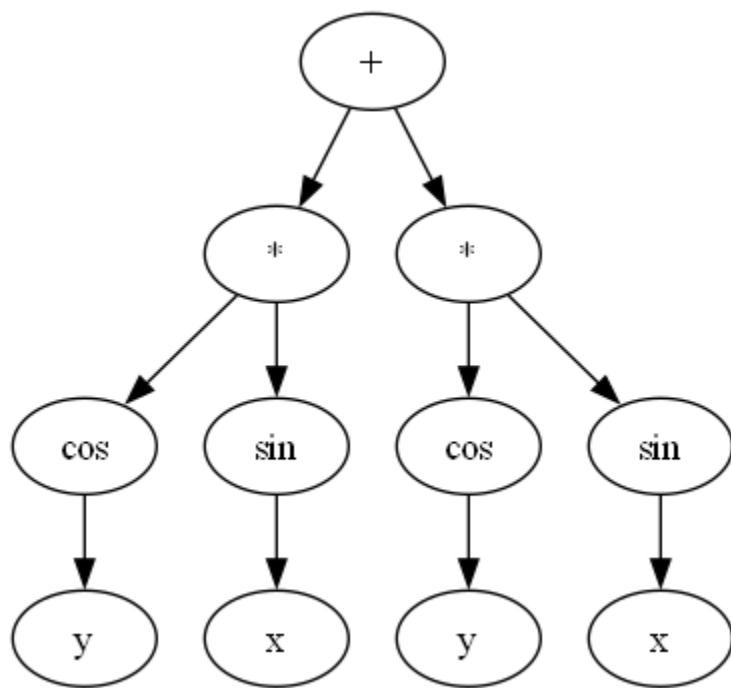
KOZA3:



NGUYEN5



NGUYEN10



Spotify regressor

<https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset/data>

dataset vnaprej očiščen in izbrane samo numerične vrednosti

'popularity', 'danceability', 'energy', 'loudness', 'speechiness', 'acousticness',
'instrumentalness', 'liveness', 'valence', 'tempo'

SAMPLE_SIZE = 5k | MAX_EVAL = 50k

Total rows loaded: 89741

Total rows for regression: 5000

Press E to pause the algorithm:

Training fitness (MSE): 405.98761721158775

Best expression:

```
((((-1.557893902651987 * ((danceability + -1.557893902651987) * (danceability -  
((5.8035614285550725 + 5.8035614285550725) + (-4.241025875362078 -  
3.099589311838283)))))) + ((danceability * ((valence + -1.557893902651987) /  
((1.0029195628934495 + -1.557893902651987) * (5.8035614285550725 *  
speechiness)))) - 3.099589311838283)) + (((5.8035614285550725 *  
(7.139500683457495 + danceability)) + ((tempo / ((7.198717190870571 + -  
1.557893902651987) * (acousticness + 7.011849124674573))) / (((danceability *  
4.9965607656368665) - tempo) * ((instrumentalness * 5.18857358285527) + (energy +  
3.3871984730366638)))))) + ((tempo / ((danceability + -1.557893902651987) *  
((instrumentalness + 5.8035614285550725) + (-4.241025875362078 -  
3.099589311838283)))))) / (((danceability * 4.9965607656368665) - 7.139500683457495)  
* (((instrumentalness / 5.8035614285550725) * (instrumentalness + -  
2.4677503259585043)) + ((5.8035614285550725 + 5.8035614285550725) + (-  
4.241025875362078 - 3.099589311838283))))))
```

Testing fitness (MSE): 424.1950273839684

SAMPLE_SIZE = 10k | MAX_EVAL = 100k

Total rows loaded: 89741

Total rows for regression: 10000

Press E to pause the algorithm:

Training fitness (MSE): 407.41093882960155

Best expression:

$$\begin{aligned} & (((((\text{danceability} + -1.4416895400831269) * (\text{acousticness} + (\text{acousticness} + \\ & (\text{acousticness} + 5.8035614285550725)))) + \text{valence}) + (((0.17555180097232714 + \\ & (((\text{acousticness} / -3.5705043167018164) - (\text{valence} + \text{speechiness})) + \\ & 5.8035614285550725)) * (7.139500683457495 + \text{danceability})) + (((\text{tempo} / (- \\ & 1.4416895400831269 / (\text{instrumentalness} + \text{speechiness}))) / (0.17555180097232714 + \\ & \text{acousticness})) / (((\text{danceability} + -1.4416895400831269) * 5.852672330457555) - \\ & ((0.17555180097232714 + \text{acousticness}) - (\text{speechiness} - -1.6299073224099399))) * \\ & (((9.693157161141471 / -5.037773211756402) / (-3.5705043167018164 * \text{danceability})) \\ & + (-5.056059397354568 + -5.923240569907726)))))) \end{aligned}$$

Testing fitness (MSE): 411.10777811681913

SAMPLE_SIZE = 10k | MAX_EVAL = 50k | novi Nodi

Total rows loaded: 89741

Total rows for regression: 10000

Press E to pause the algorithm:

Training fitness (MSE): 409.36860046405457

Best expression:

$$\begin{aligned} & ((((((\sin(\text{instrumentalness}) - \sin(9.998989200425058)) - \sqrt{(\text{instrumentalness} - \\ & \text{acousticness}))} - \sqrt{\sqrt{\sqrt{(\text{instrumentalness}))}} - \sin(\text{acousticness})) - - \\ & 9.266525924225311) * (((((\text{danceability} - \sin(\text{instrumentalness})) - \text{valence}) - \\ & \text{speechiness}) - \sqrt{(\sqrt{(\text{speechiness} - \text{acousticness})) - \text{valence}})) + \\ & 3.8188295468302)) \end{aligned}$$

Testing fitness (MSE): 411.5522855066129

SAMPLE_SIZE = 50k | MAX_EVAL = 50k |

AddNode.class, SubNode.class, MulNode.class, DivNode.class, LogNode.class

Total rows loaded: 89741

Total rows for regression: 50000

Press E to pause the algorithm:

Training fitness (MSE): 412.85029612561414

Best expression:

$$\frac{((\log(((-1.0530938789495394 / ((-0.10041526797721367 + \text{acousticness}) + (\text{instrumentalness} + -9.873499212452455)))) + (\text{speechiness} * \text{instrumentalness})))) * (\text{speechiness} + (2.149923239474763 - 8.09568854926702)))}{\log(1.45573573338684)}$$

Testing fitness (MSE): 408.8024833776891

