

Distribučovaný algoritmus diferenciální evoluce v platformě nezávislém jazyce

Ing. Petr Sklenička

- zpracování diferenciální evoluce na více výpočetních uzlech
- automatické nacházení výpočetních uzlů v síti
- dvě varianty zpracování algoritmu
- migrace jedinců mezi výpočetními uzly
- grafické znázornění vývoje nejlepšího a nejhoršího jedince
- možnost využití např. ve výuce
- využito k naučení neuronové sítě

```
public final void replaceVectorInPopulation(int indexOfVectorToReplace, double[]  
int oldVectorLength = population[indexOfVectorToReplace].length;  
int newVectorLength = newVector.length;  
if (newVectorLength == oldVectorLength) {  
    for (int i = 0; i < newVectorLength; i++) {  
        population[indexOfVectorToReplace][i] = newVector[i];  
    }  
    fitnessValues[indexOfVectorToReplace] = evaluateVector(newVector);  
} else {  
    throw new RuntimeException("The new vector has different length than  
.");  
}
```

```
ected Object[] getParametersRealValues(double[] vector) {  
Object[] realValues = new Object[vector.length];  
for (int i = 0; i < realValues.length; i++) {  
    Specimen.SpecimenParameter parameter = specimen.getParameter(i);  
    switch (parameter.getParameterType()) {  
        case REAL:  
            realValues[i] = vector[i];  
            break;  
        case INTEGER:  
            realValues[i] = (double) Math.round(  
            break;  
        case DISCRETE:  
            et  
            ath.ro
```

