Source code on JS:

```
<!DOCTYPE html>
<html>
	<head>
		<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
		<link rel="stylesheet" type="text/css" href="style.css">
		<script src="jquery-3.1.0.js"></script>
		<title>To do list</title>
	</head>
	<body>
		<div id = "table"></div>
		<script type="text/javascript">

			var statistics = [];

			function randomInit() {
				return Math.random() > 0.5 ? 1 : 0;
			}

			function sort(population) {
			return population.sort(function(first, second){
				return (first.fitness > second.fitness) - (second.fitness > first.fitness);
			});
		}

			function init() {
				var population = [];

				for(var i = 0; i < 20; i++) {
					var chromosome = [];
					for(var j = 0; j < 10; j++) {
						chromosome.push(randomInit());
```

```
                }
                population.push(chromosome);
        }
        return population;
}

function getX(chromosome) {
        var x = 0;

        for(var i = 0; i < 10; i++) {
                x += chromosome[i] ? Math.pow(2, i) / 1024 : 0;
        }
        return x;
}

function fitness(population) {
        for(var i = 0; i < population.length; i++){
                var x = getX(population[i]);
                population[i].fitness = Math.pow(Math.sin(5 * Math.PI * x), 6);
        }

        return population;
}

function d(firstChromosome, secondChromosome){
        return getX(firstChromosome) - getX(secondChromosome);
}

function uniformCrossover(firstChromosome, secondChromosome) {
        var returnChromosomes = [];
        var returnChromosome1 = [];
        var returnChromosome2 = [];
```

```
                for(var i = 0; i < 10; i++){
                        Math.random() < 0.5 ?
returnChromosome1.push(firstChromosome[i]) :
                        returnChromosome1.push(secondChromosome[i]);
                        Math.random() < 0.5 ?
returnChromosome2.push(secondChromosome[i]) :
                        returnChromosome2.push(firstChromosome[i]);
                }
                returnChromosomes.push(returnChromosome1);
                returnChromosomes.push(returnChromosome2);

                return returnChromosomes;
        }

        function mutation(population) {
                return population.map(function(chromosome) {
                        return chromosome.map(function(gene) {
                                return Math.random() < 0.02 ? (gene + 1) % 2 : gene;
                        });
                });
        }

        function nextGeneration(population) {
                var returnPopulation = population;
                var chromosomes = [];
                var statisticObj = [];

                for(var i = 0; i < 10; i++) {
                        var firstChromosomeIndex = Math.floor(Math.random() * 20);
                        var secondChromosomeIndex = Math.floor(Math.random() *
20);
```

```
                var p1 = returnPopulation[firstChromosomeIndex];
                var p2 = returnPopulation[secondChromosomeIndex];

                var children = uniformCrossover(p1, p2);
                children = fitness(children);

                var c1 = children[0];
                var c2 = children[1];

                chromosomes.push(c1.fitness);
                chromosomes.push(c2.fitness);
                chromosomes.push(p1.fitness);
                chromosomes.push(p2.fitness);

                if(d(p1, c1) + d(p2,c2) > d(p1, c2) + d(p2, c1)){
                        if(c1.fitness < p1.fitness) {
                                returnPopulation[firstChromosomeIndex] = c1;
                        }
                        if(c2.fitness < p2.fitness) {
                                returnPopulation[secondChromosomeIndex] =
c2;
                        }
                } else {
                        if(c2.fitness < p1.fitness) {
                                returnPopulation[firstChromosomeIndex] = c2;
                        }
                        if(c1.fitness < p2.fitness)          {
                                returnPopulation[secondChromosomeIndex] =
c1;
                        }
                }
        }
```

```
        statisticObj.push(chromosomes);
        statisticObj.push(returnPopulation);
        statisticObj.push(averageFitness(returnPopulation));
        statistics.push(statisticObj);
        return returnPopulation;
}

function averageFitness(population){
        var average = 0;
        for(var i = 0; i < population.length; i++){
                average += population[i].fitness;
        }

        return average / population.length;
}

var population = init();
population = fitness(population);

for(var i = 0; i < 15; i++){
        population = nextGeneration(population);
}

function show(population) {
        for(var i = 0; i < 20; i++){
                console.log(getX(population[i]));
        }
}

function showStatistics(statistics){
        for(var i = 0; i < 15; i++){
```

```
                    if(i % 3 === 0){
                        for(var j = 0; j < 10; j++){
                            console.log("c1: " + statistics[i][0][j * 4].toFixed(5) + " \tc2: " + statistics[i][0][j * 4 + 1].toFixed(5) + "\np1: " + statistics[i][0][j * 4 + 2].toFixed(5) + "\t\tp2: " + statistics[i][0][j * 4 + 3].toFixed(5));
                        }
                    }
                }

                for(var i = 0; i < 15; i++){
                    if(i % 3 === 0){
                        for(var k = 0; k < 20; k++){
                            console.log("X:" + getX(statistics[i][1][k]).toFixed(7) + "\t\tY: " + statistics[i][1][k].fitness.toFixed(7));
                        }
                    }
                }

                console.log("average: \n");
                for(var i = 0; i < 15; i++){
                    console.log(statistics[i][2]);
                }
                console.log("min: \n");
                for(var i = 0; i < 15; i++){
                    console.log(sort(statistics[i][1])[0].fitness);
                }
            }

            for(var i = 0; i < population.length; i++){
                console.log(getX(population[i]));
            }

            //showStatistics(statistics);
```

```
        </script>
    </body>
</html>
```