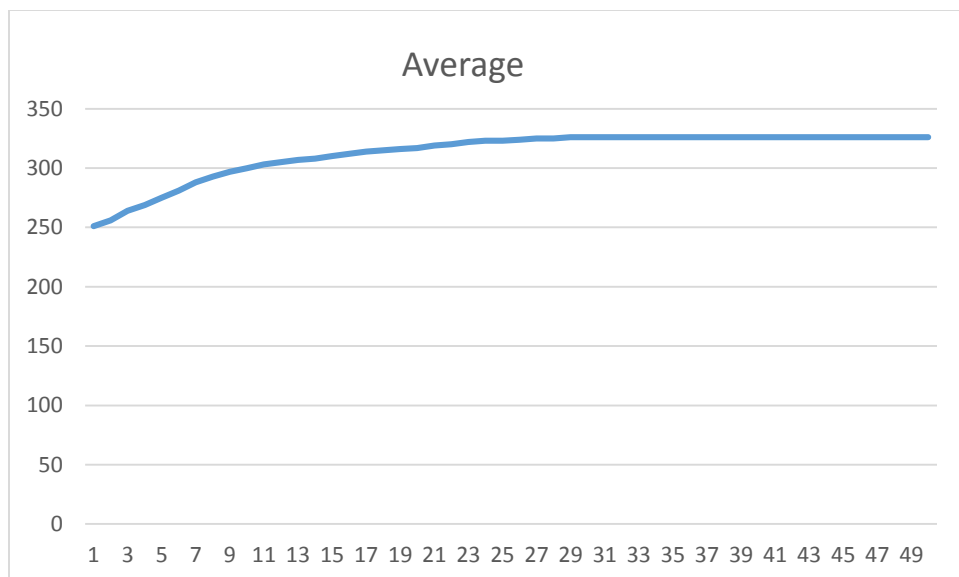
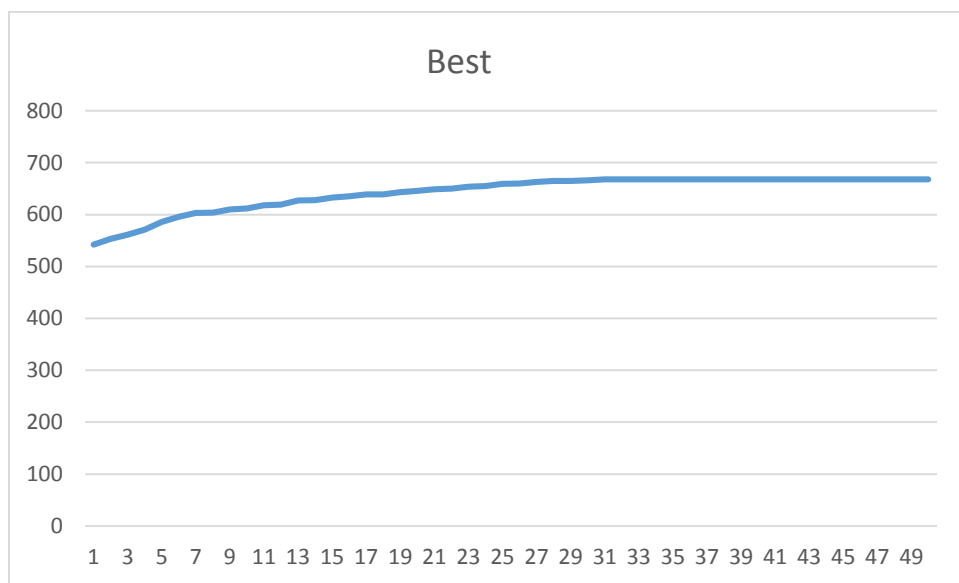


Modern intelligent IT
Lab 1 (26.02.2016)
Akira Imada
Student – Dzmitry Rybalko(AI - 10)

All one problem

We created a random genes and over 50 iterations randomly mated individuals from the best half of the generation. The results of these operations can be seen on charts



Code of program:

```
((() => {
  "use strict";

  const populationCount = 100,
        genSize = 1000,
        iterationsCount = 50;

  let generation = generateGeneration(populationCount, genSize),
      iterationNum = 0;

  for (let i = 0; ++i < iterationsCount;) {
    generation = generation.sort((gen1, gen2) => {
      return calcOnes(gen2) - calcOnes(gen1);
    });
    logGeneration(generation);

    generation = nextGeneration(generation);
  }

  logGeneration(generation);
  return;

  function generateGeneration(count, size) {
    let generation = [];
    for (let i = -1; ++i < count;) {
      let gen = [];
      for (let j = -1; ++j < size;) {
        gen.push(getRandomBinary());
      }
      generation.push(gen);
    }
    return generation;
  }

  function calcOnes(gen) {
    let count = 0;
    for (let i = -1; ++i < gen.length;) if (gen[i]) count++;
    return count;
  }

  function nextGeneration(parentGeneration) {
    let repeatCount = populationCount / 2,
        newGeneration = [];
    for (let i = 0; ++i < populationCount;) {
      let firstIndex = getRandomInt(0, repeatCount),
          secondIndex = getRandomInt(0, repeatCount);
      newGeneration.push(...crossover(parentGeneration[firstIndex],
parentGeneration[secondIndex]));
    }
    return newGeneration;
  }
}
```

```

    }

    function crossover(firstParent, secondParent) {
        let crossIndex = getRandomInt(1, firstParent.length);
        let firstPartFirst = firstParent.slice(0, crossIndex),
            secondPartFirst = firstParent.slice(crossIndex, firstParent.length),
            firstPartSecond = secondParent.slice(0, crossIndex),
            secondPartSecond = secondParent.slice(crossIndex, secondParent.length);

        return [firstPartFirst.concat(secondPartSecond),
            firstPartSecond.concat(secondPartFirst)];
    }

    function getRandomBinary(min, max) {
        return Math.round(Math.random());
    }

    function getRandomInt(min, max) {
        return Math.floor(Math.random() * (max - min)) + min;
    }

    function logGeneration(generation) {
        let average = 0;
        for (let i = -1; ++i < generation.length; i++) average +=
            calcOnes(generation[i]);
        average = Math.floor(average / generation.length);
        console.log(average);
    }
}());

```