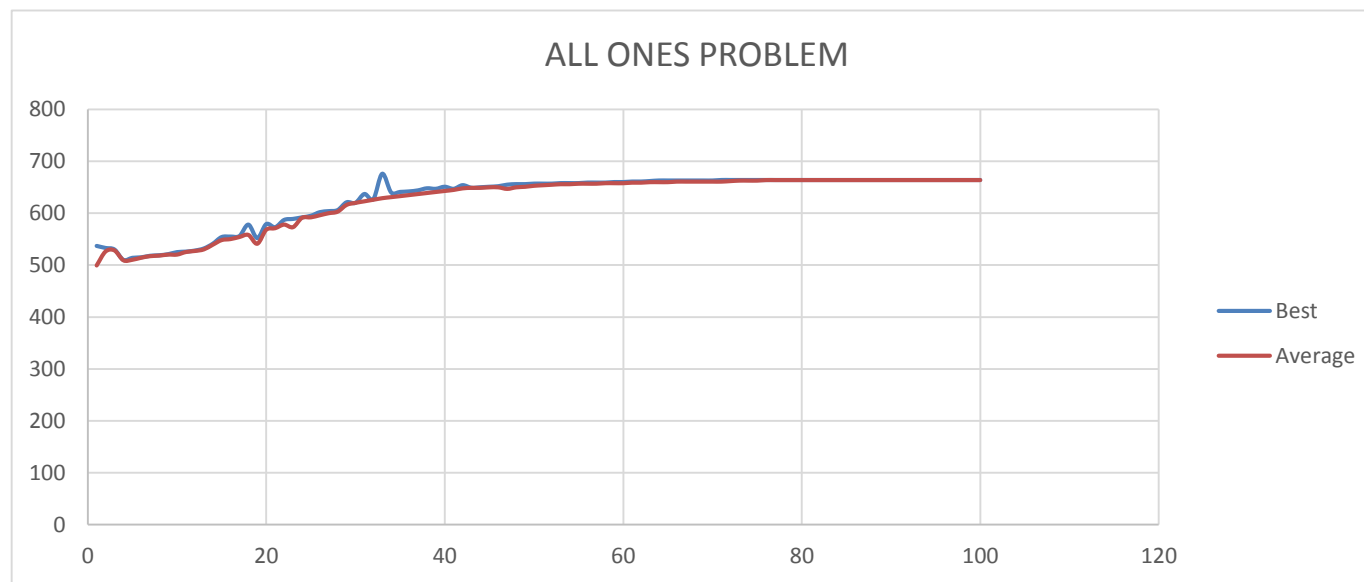


Task 1 All One Problem (12.02.2016)

Student –Artyom Yudenkov

We have the initial population with 100 binary chromosomes that are with 1000 genes. The fitness under the number of “1” is good, “0” - bad. We need to get gene which consists of all “1”. For these Achievement we use such algorithm :

1. Create 100 binary-chromosomes at random where each chromosome contains 1000 gens.
2. Select 2 chromosomes with Roulette method.
3. With one-point-crossover create new generation of children.
4. Repeat steps 2-3, until we get new population.
5. Repeat 4. until the fitness value does not change any more.



No of iter.	FITNESS	CHANCE	CHOICE
1	537	0,010752	2

535	0,010662	0
532	0,010652	0

528	0,010572	0
528	0,010572	0

526	0,010532	1
526	0,010532	2
524	0,010492	1
523	0,010472	1
522	0,010452	2
520	0,010412	1
520	0,010412	2
519	0,010392	0
519	0,010392	0
517	0,010352	3
517	0,010352	0
517	0,010352	2
517	0,010352	1
515	0,010312	1
515	0,010312	0
513	0,010272	2
513	0,010272	1
513	0,010272	1
513	0,010272	2
512	0,010251	1
511	0,010231	1
511	0,010231	0
511	0,010231	0
510	0,010211	0
509	0,010191	1
509	0,010191	1
509	0,010191	2
508	0,010166	2
507	0,010151	2

505	0,010111	2
504	0,010091	1
504	0,010091	1
503	0,010066	0
503	0,010066	0
503	0,010066	0
503	0,010066	0
502	0,010051	0
502	0,010051	0
502	0,010051	0
501	0,010031	0
500	0,010011	1
500	0,010011	1
500	0,010011	2
500	0,010011	2
500	0,010011	1
499	0,009991	2
499	0,009991	2
499	0,009991	0
498	0,009966	1
498	0,009966	2
497	0,009951	2
496	0,009931	0
496	0,009931	2
496	0,009931	1
495	0,009911	2
495	0,009911	2
494	0,009891	1
494	0,009891	0

494	0,009891	0
493	0,009866	2
493	0,009866	2
493	0,009866	2
493	0,009866	2
492	0,009851	2
492	0,009851	0
492	0,009851	1
492	0,009851	2
491	0,009831	1
490	0,009811	2
490	0,009811	1
490	0,009811	2
489	0,009791	1
489	0,009791	2
489	0,009791	1
487	0,009751	2
487	0,009751	0
484	0,009691	0
483	0,009666	1
482	0,009651	1
481	0,009631	0
480	0,009611	1
480	0,009611	0
479	0,009591	1
479	0,009591	1
479	0,009591	1
478	0,009566	0

	478	0,009566	0
	477	0,009551	0
	476	0,009531	0
	470	0,009411	1
	470	0,009411	1
	468	0,009365	1
	463	0,009265	1
	459	0,00919	1
	448	0,008965	1
	sum	0,999865	100
25	641	0,0101	1
	640	0,01	3
	640	0,01	1
	640	0,01	1
	640	0,01	0
	640	0,01	0
	640	0,01	0
	640	0,01	0
	640	0,01	0
	640	0,01	0
	640	0,01	2
	640	0,01	1
	640	0,01	1
	640	0,01	2
	639	0,01	1
	639	0,01	0
	639	0,01	1

639	0,01	1
639	0,01	1
639	0,01	1
639	0,01	0
639	0,01	0
639	0,01	2
639	0,01	1
639	0,01	2
638	0,01	1
638	0,01	1
638	0,01	1
638	0,01	2
638	0,01	2
638	0,01	1
638	0,01	0
638	0,01	1
638	0,01	0
638	0,01	0
638	0,01	1
638	0,01	1
638	0,01	2
638	0,01	1
638	0,01	1
638	0,01	2
638	0,01	2
638	0,01	3
638	0,01	0
638	0,01	0
638	0,01	0

638	0,01	0
638	0,01	0
638	0,01	0
638	0,01	1
638	0,01	1
638	0,01	1
638	0,01	0
638	0,01	1
637	0,01	2
637	0,01	2
637	0,01	1
637	0,01	1
637	0,01	1
637	0,01	0
637	0,01	2
636	0,01	1
636	0,01	1
636	0,01	2
636	0,01	2
636	0,01	1
636	0,01	1
636	0,01	1
636	0,01	0
636	0,01	2
636	0,01	0
636	0,01	0
636	0,01	0

664	0,01	1
664	0,01	0
664	0,01	2
664	0,01	1

664	0,01	2
664	0,01	2
664	0,01	1
664	0,01	1

	664	0,01	1
	664	0,01	1
	sum	1	100

Code:

```

<html>
<body>
<br>
<br>
<script language="JavaScript">
  (() => {
"use strict";
const populationCount = 100,
genSize = 1000,
ittrationsCount = 100;
let generation = RouletteGenerateGeneration(populationCount, genSize),
ittrationNum = 0;
for (let i = 0; ++i < ittrationsCount;) {
  generation = generation.sort((gen1, gen2) => {
return calcOnes(gen2) - calcOnes(gen1);
});
logGeneration(generation);
generation = nextGeneration(generation);
}
logGeneration(generation);
return;
function RouletteGenerateGeneration(count, size) {
let generation = [], chance=0;
for (let i = -1; ++i < count;) {
let gen = [],forevery (chance) {insert onlyOnesCh;}
for (let j = -1; ++j < size;) {
gen.push(getRandomBinary());
forevery(chance) {where top value(chance) insert parentsOne;}
for ( int i=0; i<sets.size();i++){
sum+= sets[i].eval();
}
}
}

```

```

    rand = ((rand() / RAND_MAX) * sum);
    sum=0;
    for ( int i =0; i<sets.size(); i++){
        sum+= sets[i].eval();
        if(rand<sum){ break;
        }}
    }
    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, populationCount),
            secondIndex = getRandomInt(0, populationCount);
        document.write(firstIndex + " " + secondIndex + " | ");
        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 bestCh (generation) {
let best = calcOnes(generation[0]); ;
  for (let i = 1; i < generation.length; i++) {
    if (calcOnes(generation[i])>best) best = calcOnes(generation[i]);
  }
return best;
}
function logGeneration(generation) {
let average = 0;
let allsum=0;
let onlyOnes=0;
let  onlyOnesCh=0;
for (let i = 0; i < generation.length; i++) allsum +=
calcOnes(generation[i]);
  for (let i = 0; i < generation.length; i++)
  {onlyOnes=calcOnes(generation[i]);
  onlyOnesCh=onlyOnes/allsum;
  document.write(onlyOnes+"|" + onlyOnesCh + '<br>');}
```

```
average = Math.floor(allsum / generation.length);
document.write("Average "+average + '<br>');
document.write("Best" + bestCh(generation)+'<br>');
```

```
}
```

```
})();
</script>
<br>
</body>
</html>
```