Project 4
Search Based Testing

Task

Generate test inputs that achieve full branch coverage.

Example

```
public class Example {
int a;
static void m(int x, int y) {
        if (x + y > 10) {
            a = 1;
        }
    }
}
```

Inputs:

5,6

0,0

Genetic Algorithm

- See lecture on search-based testing

Search-based Testing

Software Engineering

Gordon Fraser • Saarland University

Start

- Choose target to cover.
- Generate start population.

Example

Target to cover:
Condition in if evaluates to false

Start Population:

13,4

7,8

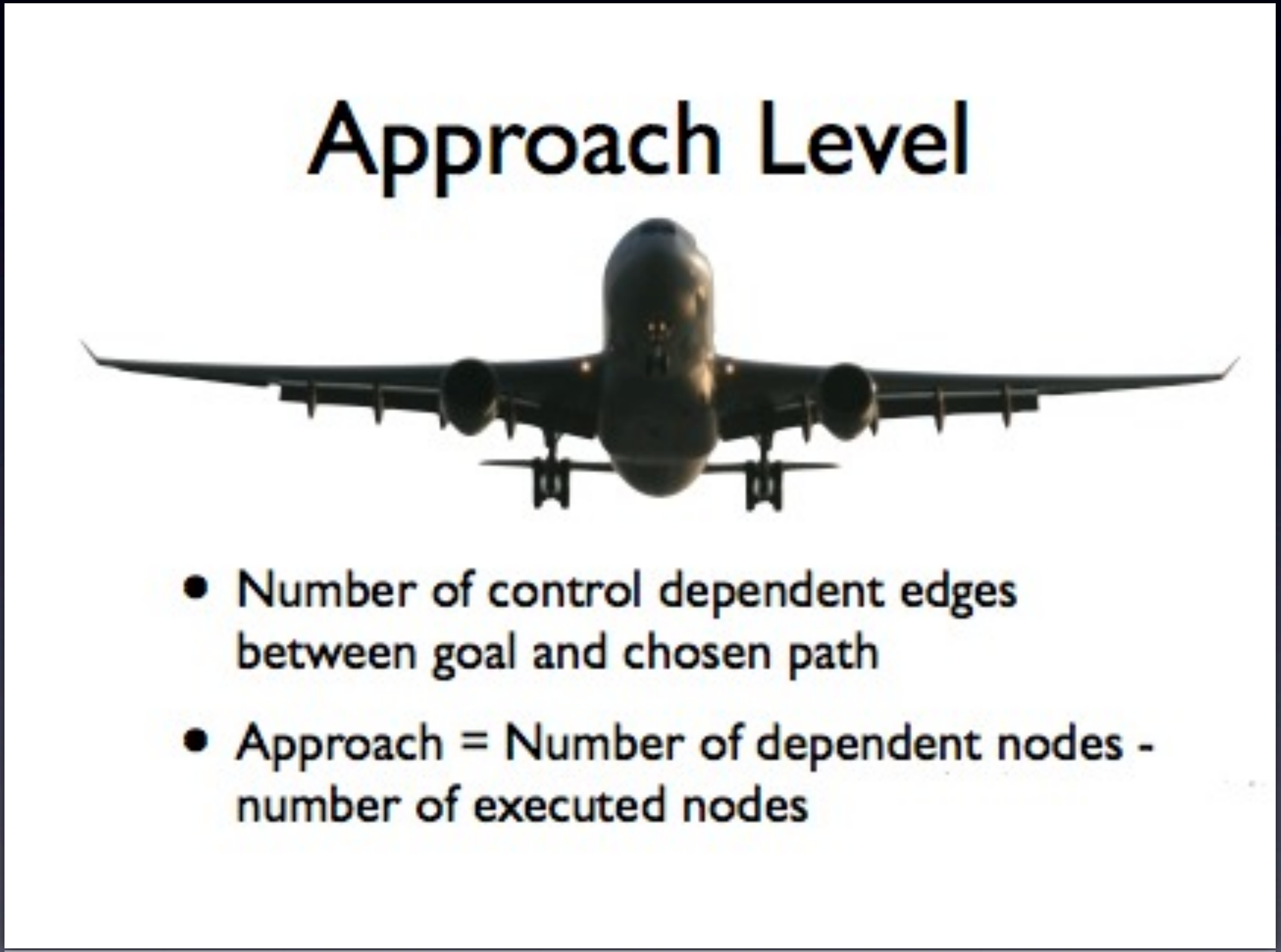
2,10

Compute Fitness

- Fitness:

  approach level + normalized branch distance

Approach Level



Approach Level

- Number of control dependent edges between goal and chosen path
- Approach = Number of dependent nodes - number of executed nodes

Example

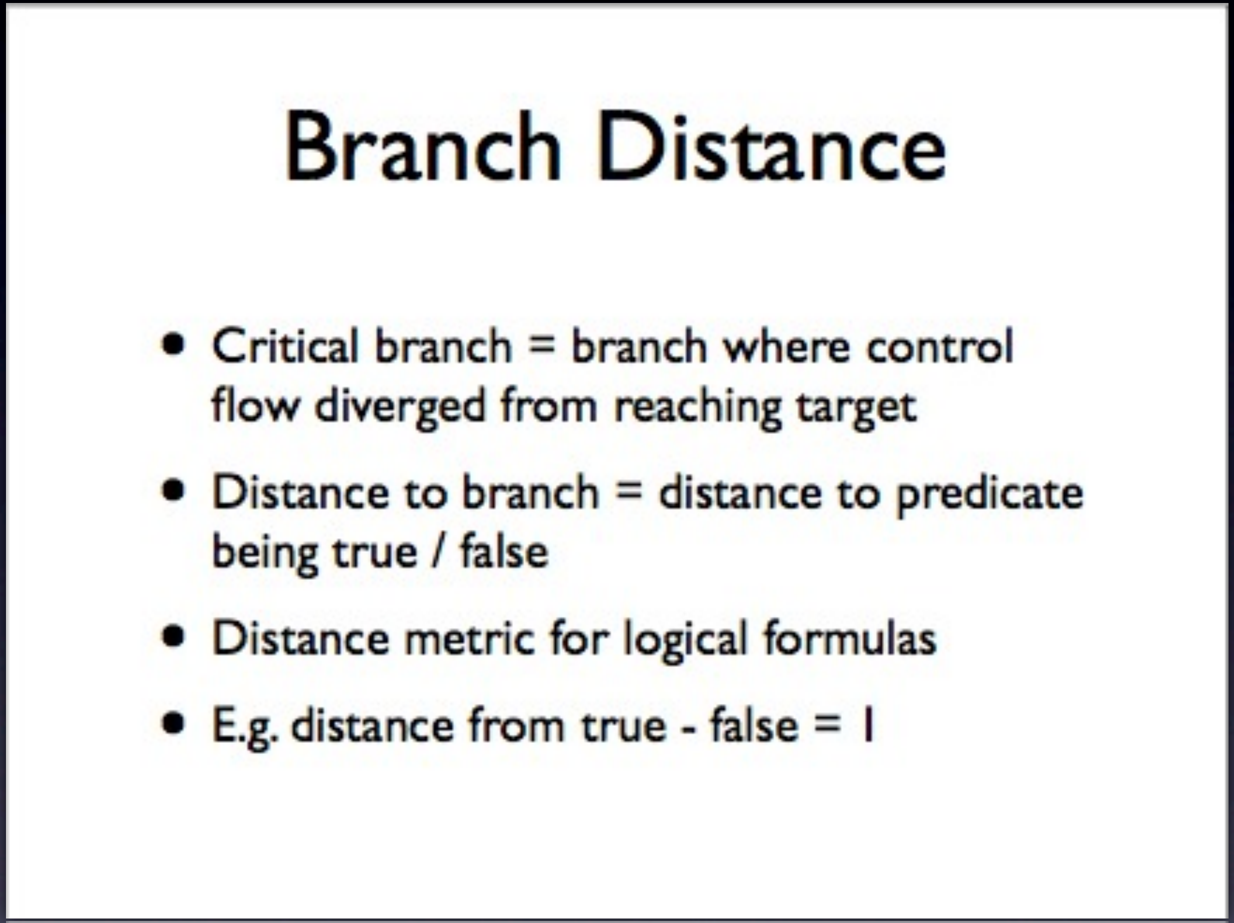
```
public class Example {
int a;
static void m(int x, int y) {
        if (x + y > 10) {
            a = 1;
        }
    }
}
```

Inputs:

13,4 7,8 2,10

Approach Level: 0

Distance



Branch Distance

- Critical branch = branch where control flow diverged from reaching target
- Distance to branch = distance to predicate being true / false
- Distance metric for logical formulas
- E.g. distance from true - false = 1

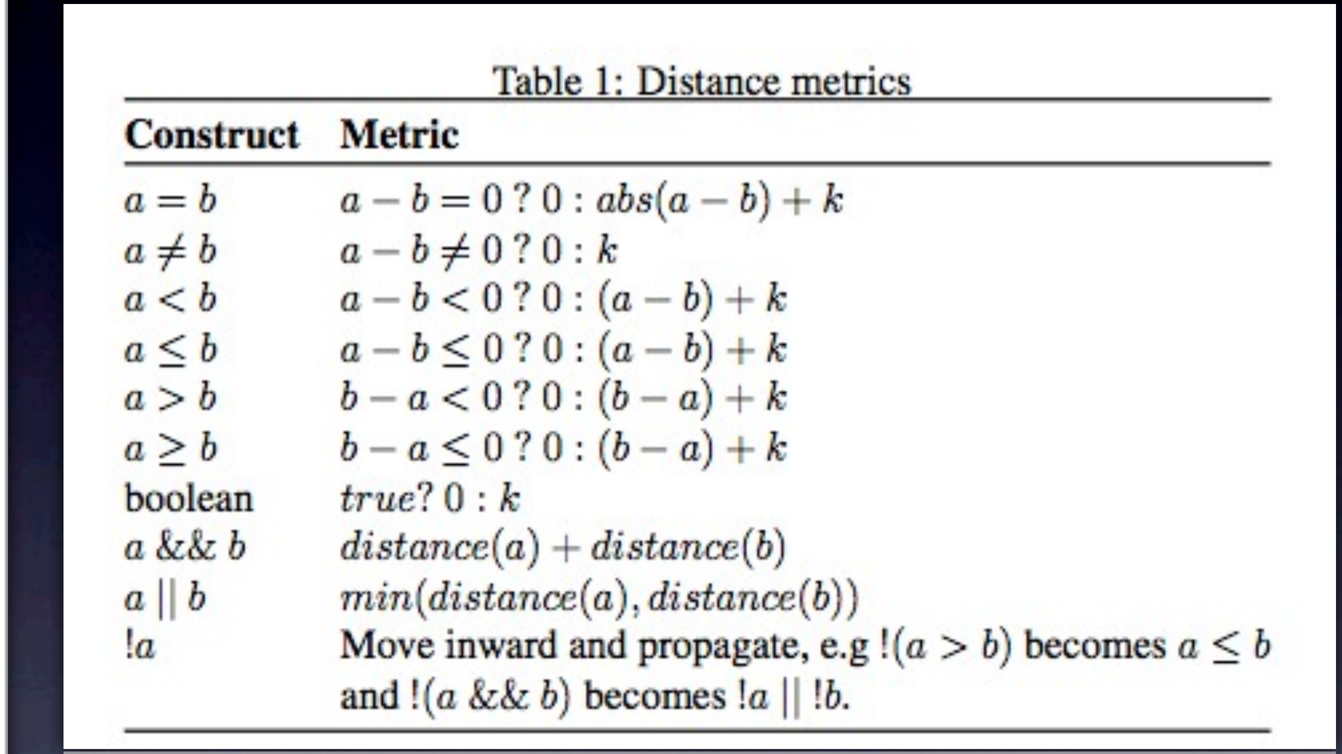


Table 1: Distance metrics

Construct	Metric
$a = b$	$a - b = 0 \,?\, 0 : abs(a - b) + k$
$a \neq b$	$a - b \neq 0 \,?\, 0 : k$
$a < b$	$a - b < 0 \,?\, 0 : (a - b) + k$
$a \leq b$	$a - b \leq 0 \,?\, 0 : (a - b) + k$
$a > b$	$b - a < 0 \,?\, 0 : (b - a) + k$
$a \geq b$	$b - a \leq 0 \,?\, 0 : (b - a) + k$
boolean	$true? \, 0 : k$
$a \,\&\&\, b$	$distance(a) + distance(b)$
$a \,\|\|\, b$	$min(distance(a), distance(b))$
$!a$	Move inward and propagate, e.g $!(a > b)$ becomes $a \leq b$ and $!(a \,\&\&\, b)$ becomes $!a \,\|\|\, !b$.

Example

```
public class Example {
int a;
static void m(int x, int y) {
        if (x + y > 10) {
            a = 1;
        }
    }
}
```

Inputs:	Distance:
13,4	17 -10 + k
7,8	15 -10 + k
2,10	12 -10 + k

Instrumentation

```
static void m(int x, int y) {
traceDist(10 - (x+y)<   0 ? 0 : (10-(x+y)) + k, 0);
traceDist((x+y) - 10<= 0 ? 0 : (x+y-10) + k, 1);
        if (x + y > 10) {
            a = 1;
        }
    }
```

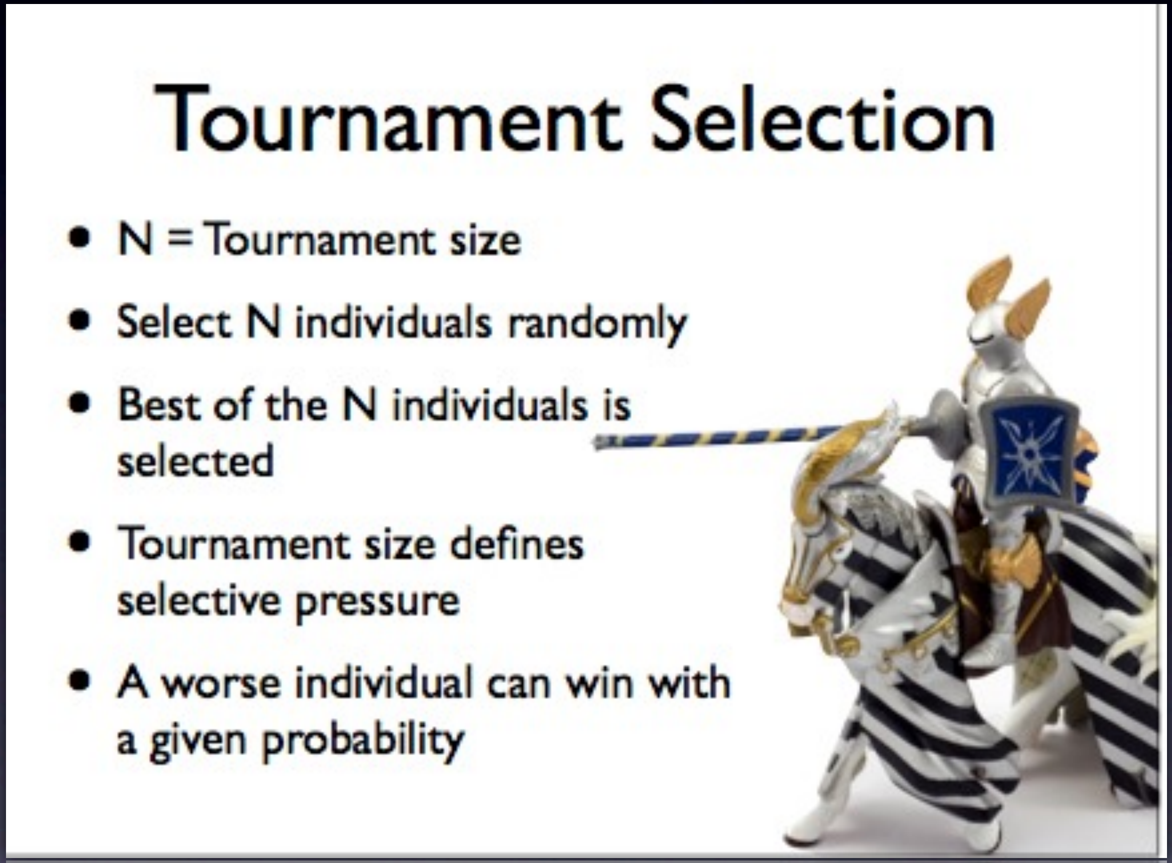
Fitness

Inputs:	Fitness:
13,4	0 + 8/9
7,8	0+6/7
2,10	0+3/4

Elitism

- Keep best chromosomes. For next generation.
- In example: 2,10

Tournament Selection

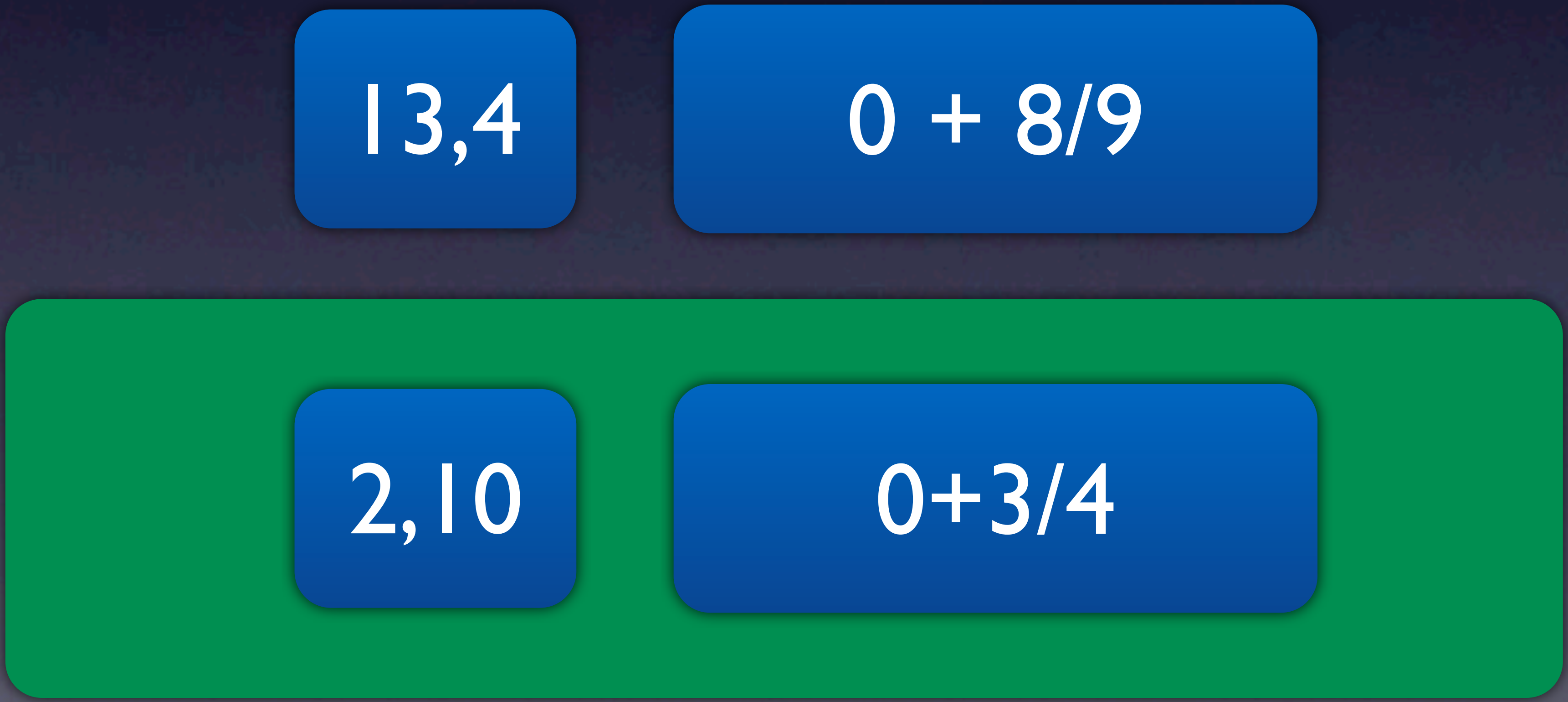


Tournament Selection

- N = Tournament size
- Select N individuals randomly
- Best of the N individuals is selected
- Tournament size defines selective pressure
- A worse individual can win with a given probability

Example

Tournament Size: 2



13,4

0 + 8/9

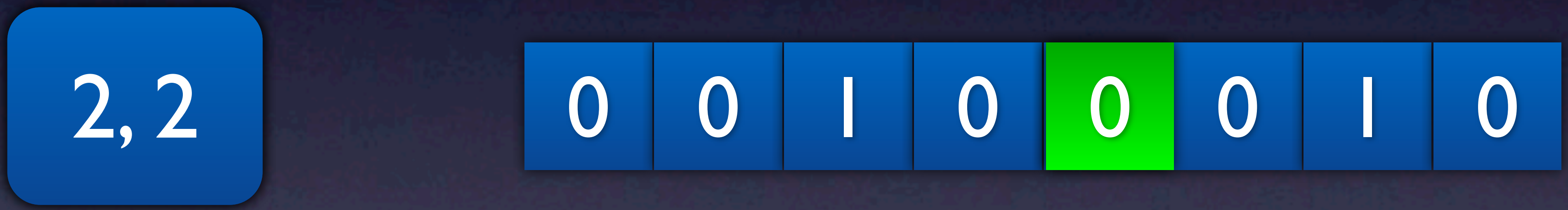
2,10

0+3/4

Crossover

3, 8	0	0	1	0	1	0	1	0
6,10	0	1	1	1	1	0	0	0

Mutation



2, 2

0	0	1	0	0	0	1	0

Stopping condition

- Maximum number of iterations reached.
- Target is covered. (fitness is 0)
- Write out generated inputs to csv-file.
- Else repeat the steps (elitism, crossover, mutation).