Automatic Algorithm Invention

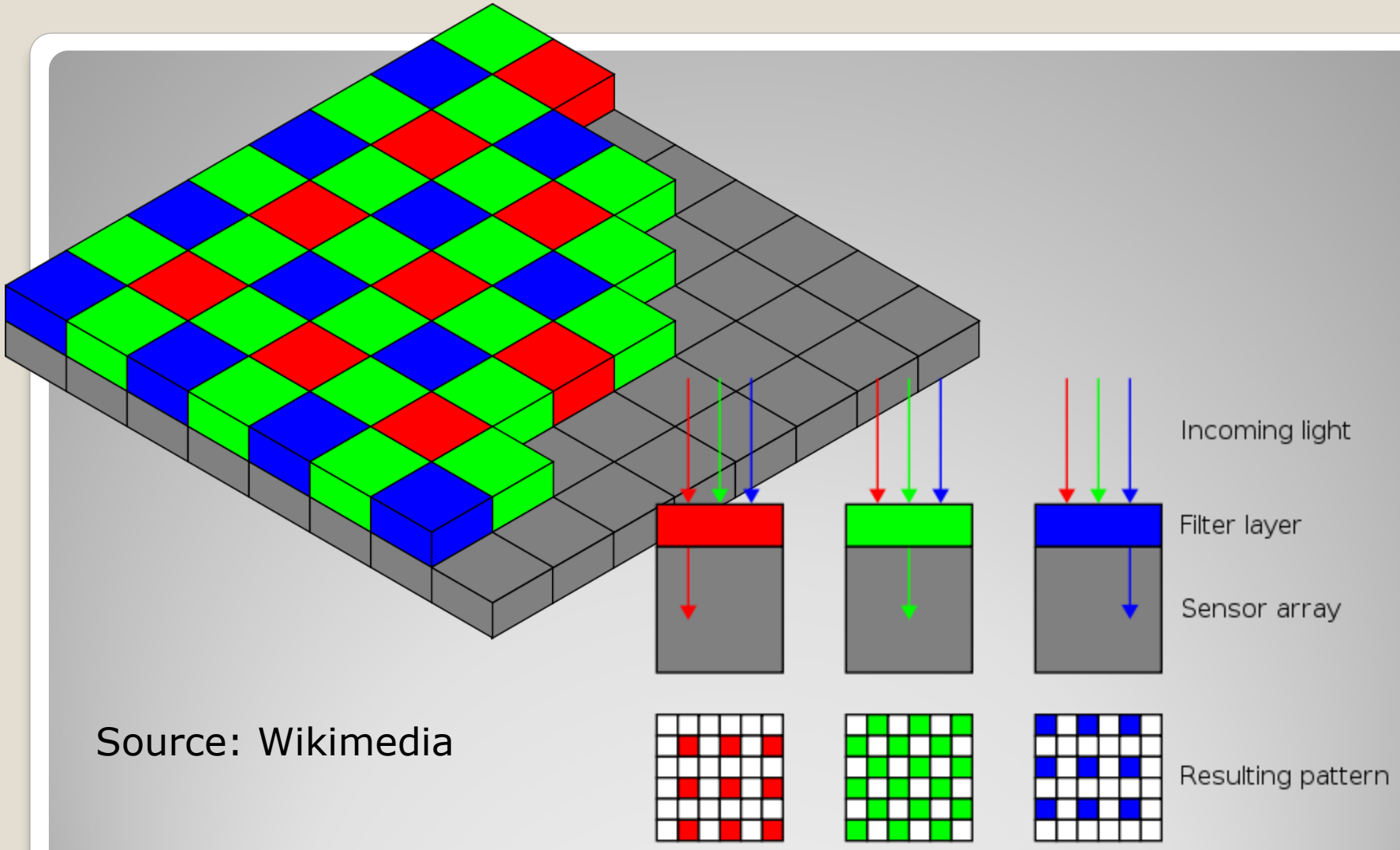
by Wes Faler of Part-Time Scientists
for 28C3, Berlin Germany, 2011
wf@ptscientists.com

- Why
- What
- How
- Code and Details
- Scaling
- State
- Your turn!

<presentation>

Why

- Part Time Scientists needs a De-Bayer algorithm for our images.



Source: Wikimedia

What - Bayer Image Filter

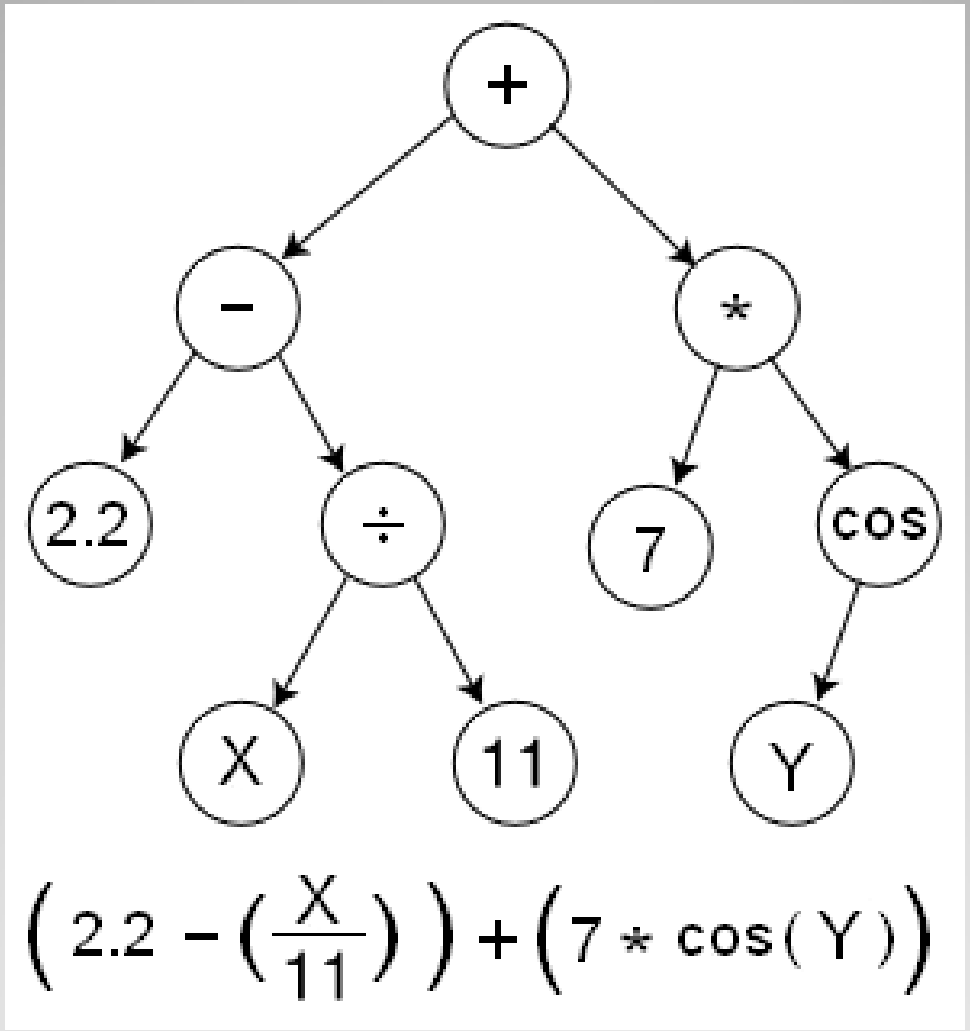
- Given:
  - An existing system structure
    - Inputs
    - Output(s)
    - Formula structure
  - Test cases
  - Constraints
- Create:
  - The optimal set of parameters

$$a\ Red + b\ Blue + c\ Green_0 + d\ Green_1$$

Optimization – GA, PS, SA, LS

- Given:
  - A partial existing system structure
    - Inputs
    - Output(s)
    - ~~Formula structure~~
  - Test cases
  - Constraints
- Create:
  - ~~The optimal set of parameters~~
  - An equation or algorithm

Invention – GP



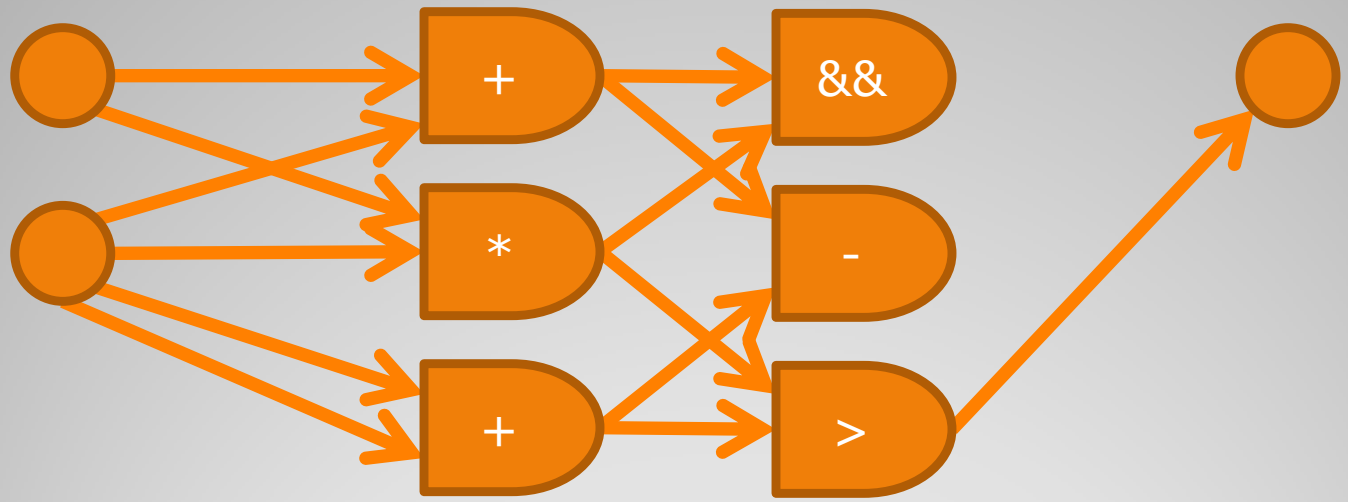
Invention - GP

Source: Wikimedia

How – CGP

- Cartesian Genetic Programming (CGP)
  - Generates equations like circuits.
  - Parallelizable results.
  - FPGA friendly.

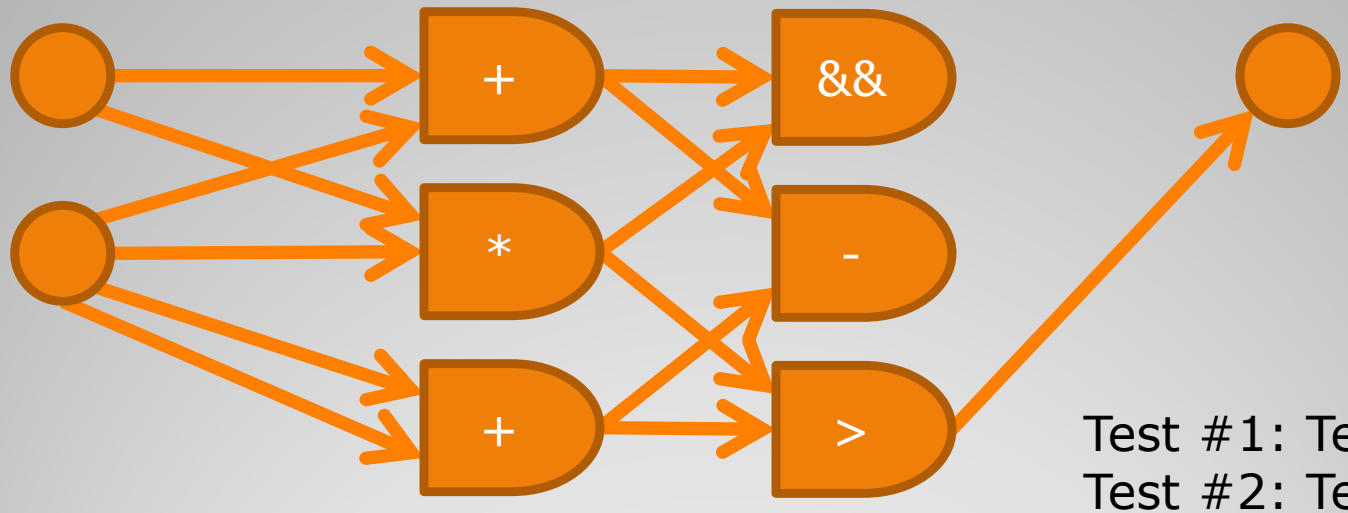
- Make a random “circuit”.
- Grid layout.



How – CGP

How – CGP

- Score the circuit.

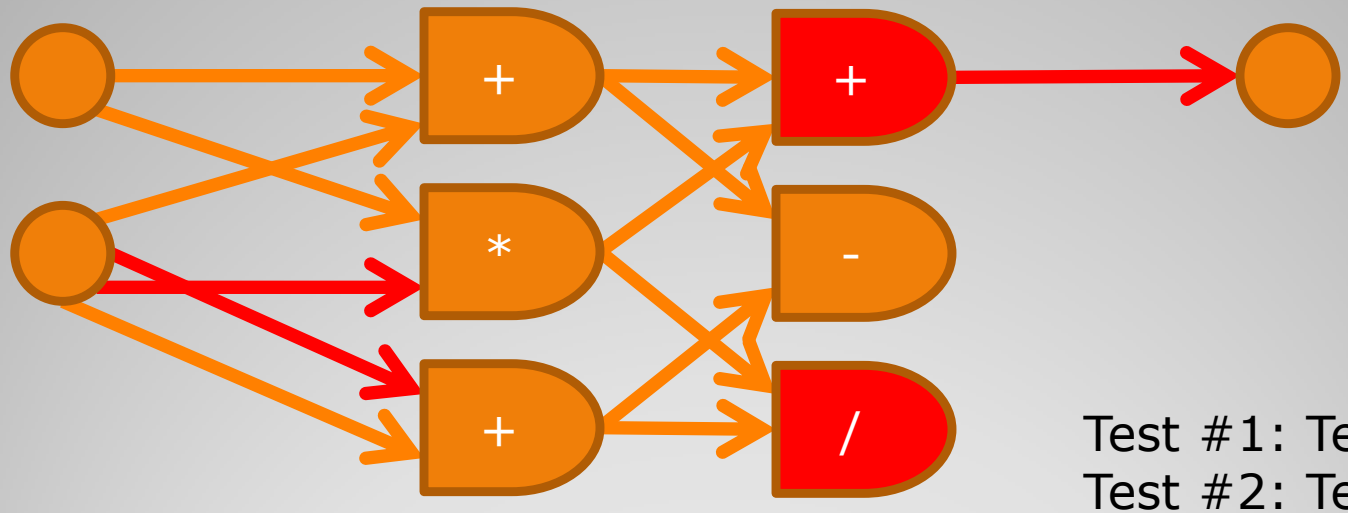


Test #1: Terrible
Test #2: Terrible
Test #3: Terrible

Score: Terrible*3

How – CGP

- Make random changes and rescore.



Test #1: Terrible
Test #2: Terrible-4
Test #3: Terrible

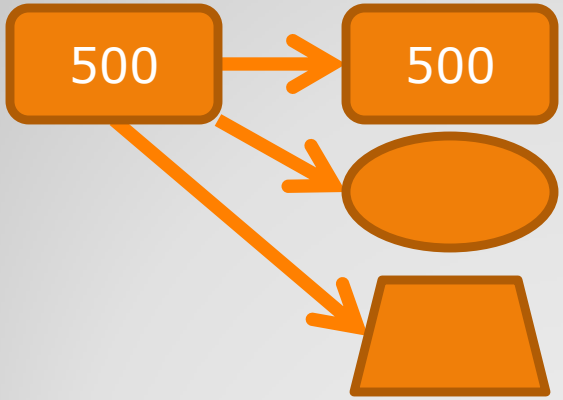
Score: Terrible*3-4

How - CGP

- Start with 1 parent “circuit”.

500

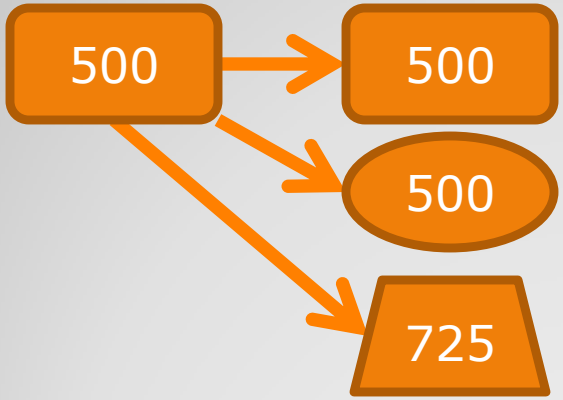
- Start with 1 parent “circuit”.
- Make mutant children.



How - CGP

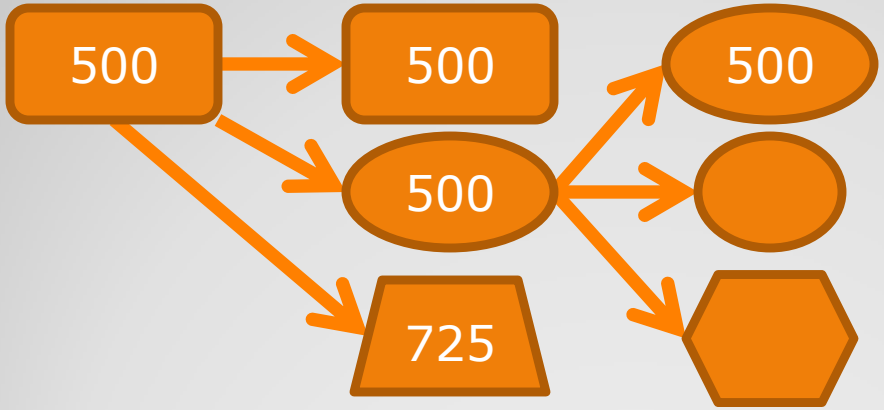
How - CGP

- Start with 1 parent “circuit”.
- Make mutant children.
- Score everyone.



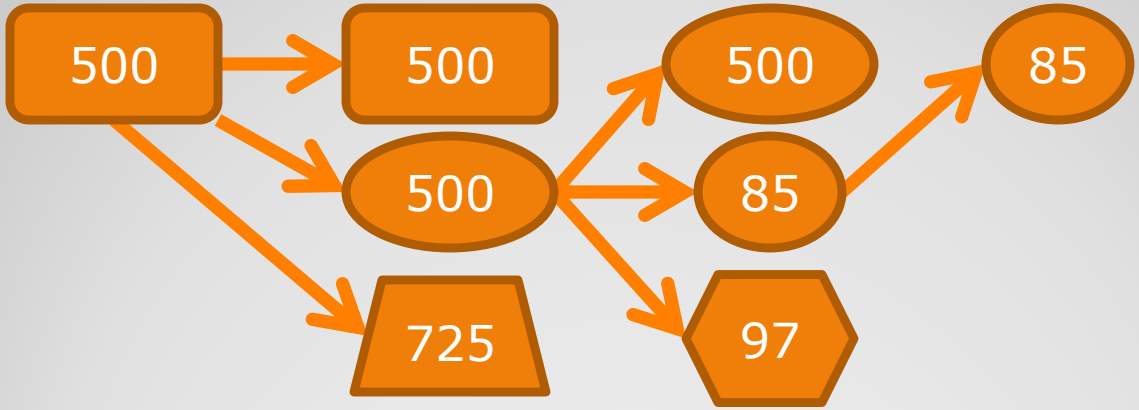
How - CGP

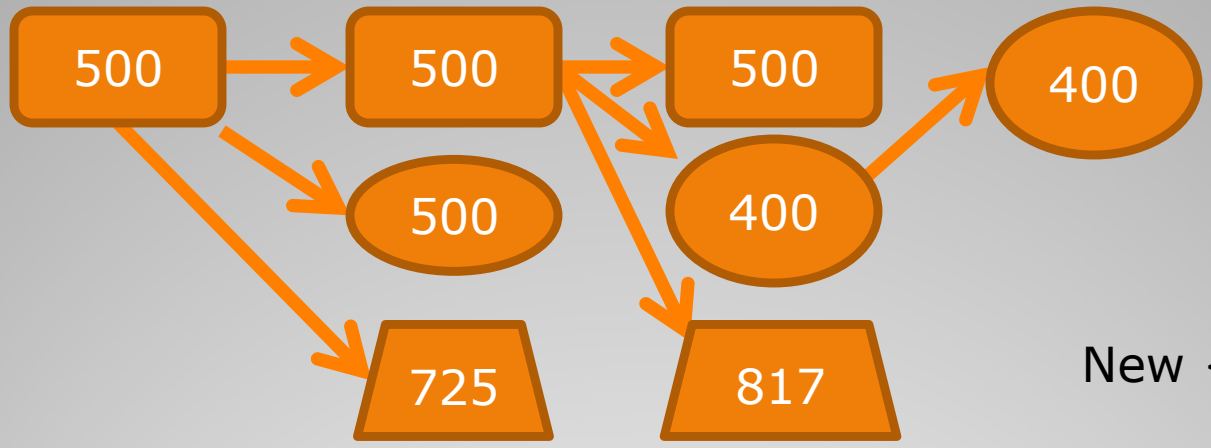
- Start with 1 parent “circuit”.
- Make mutant children.
- Score everyone.
- Promote the best child that isn’t worse than the parent.



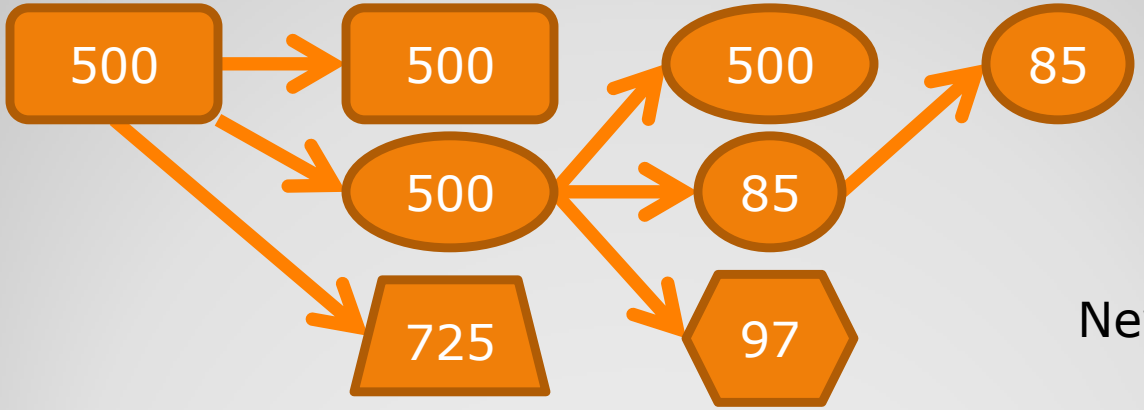
How - CGP

- Start with 1 parent “circuit”.
- Make mutant children.
- Score everyone.
- Promote the best child that isn’t worse than the parent.



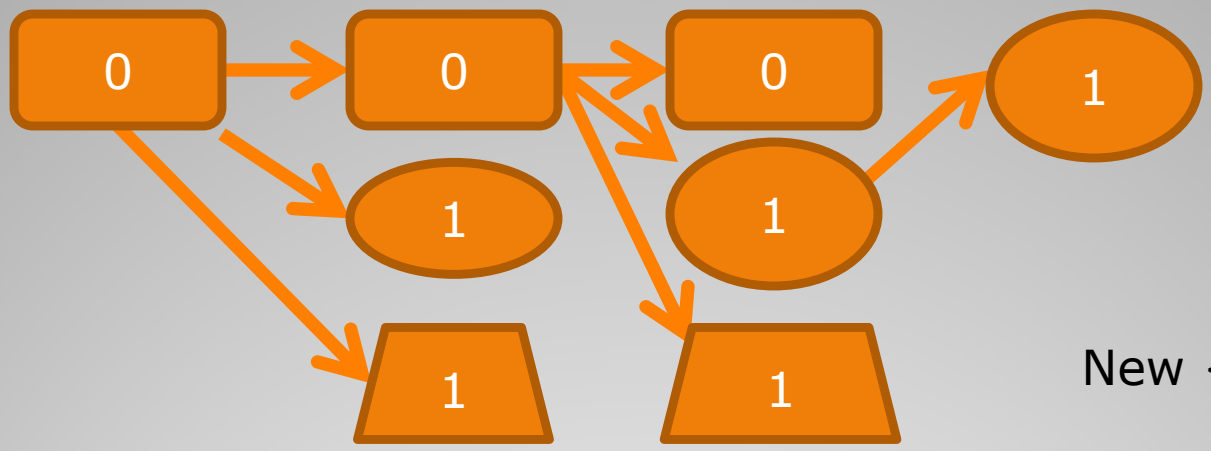


New < Old

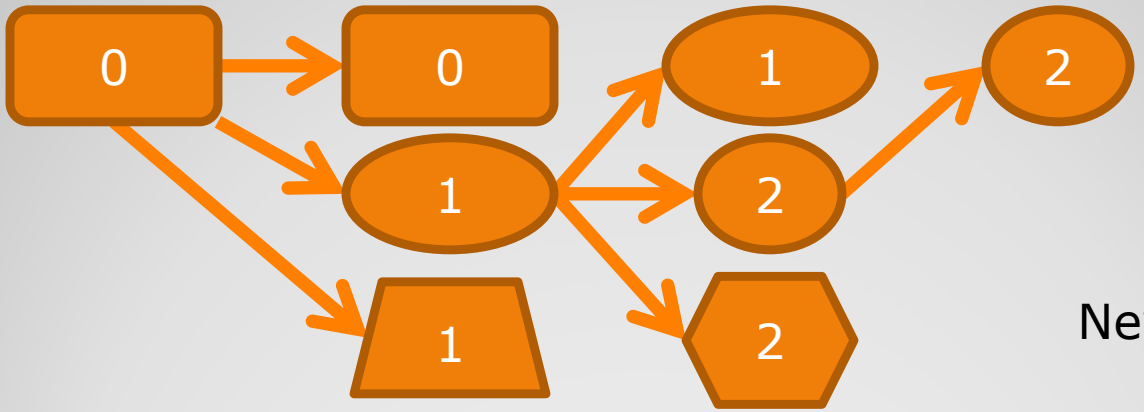


New <= Old

Neutral Search - Scores



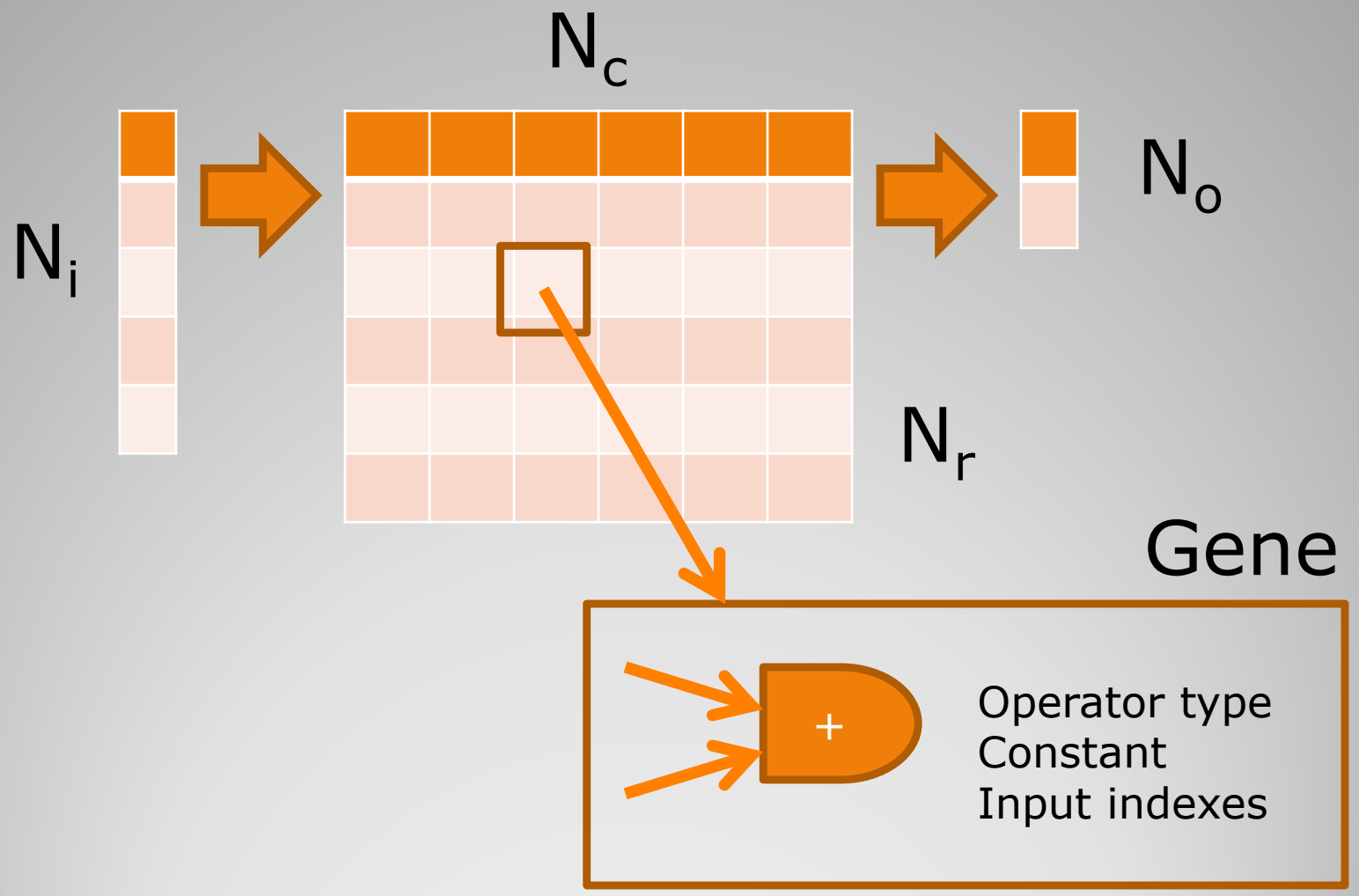
New < Old



New <= Old

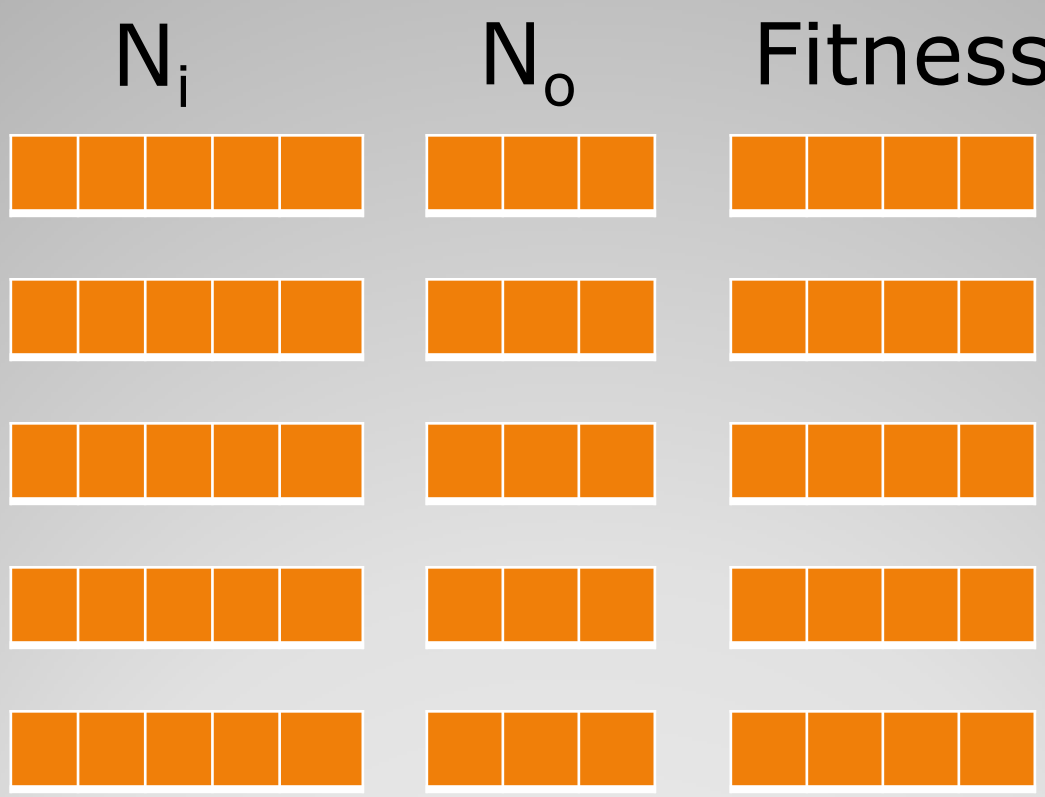
Neutral Search – Mutations

Terminology



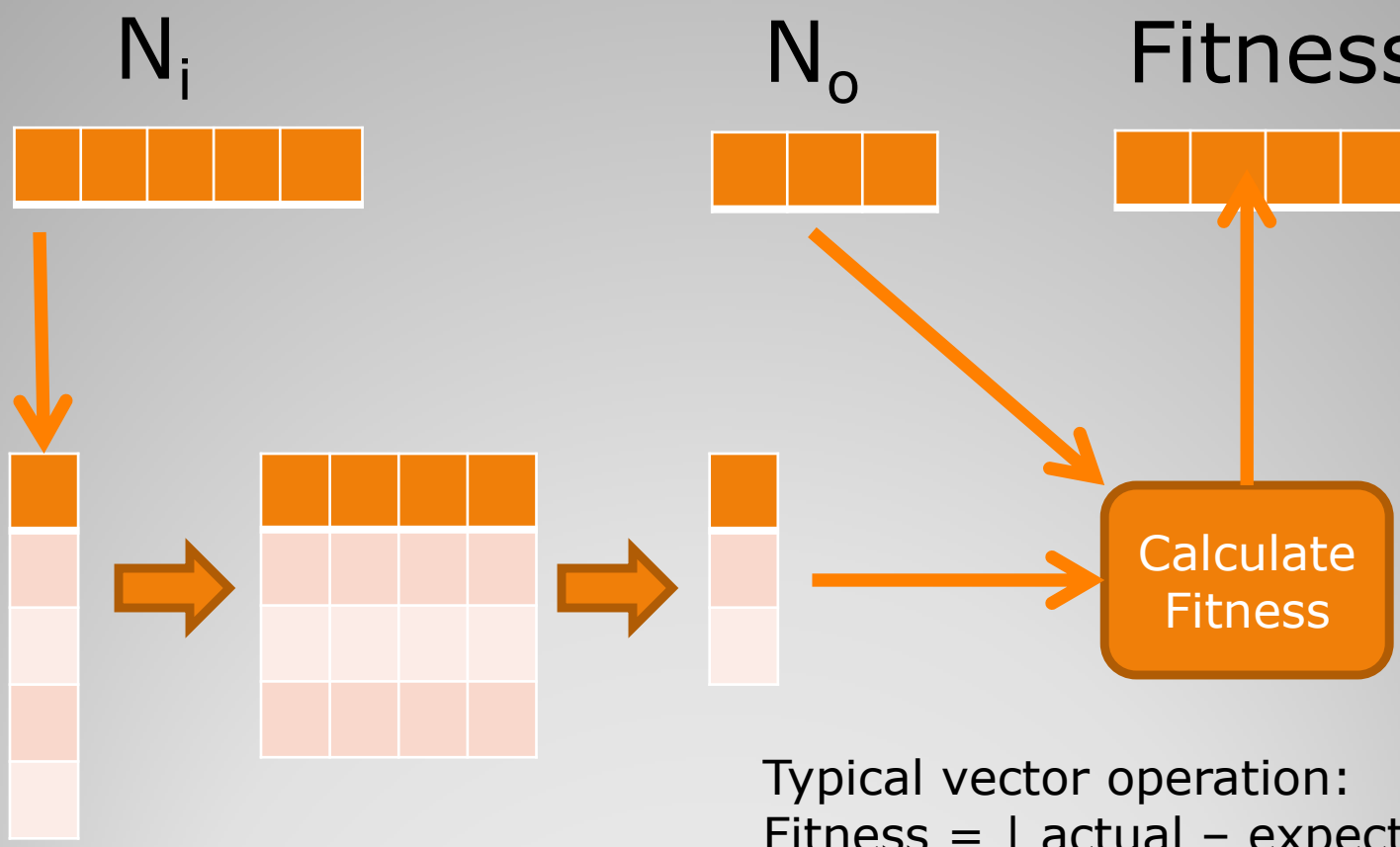
- Individual
  - Representation
  - Random creation
- Execution

Code

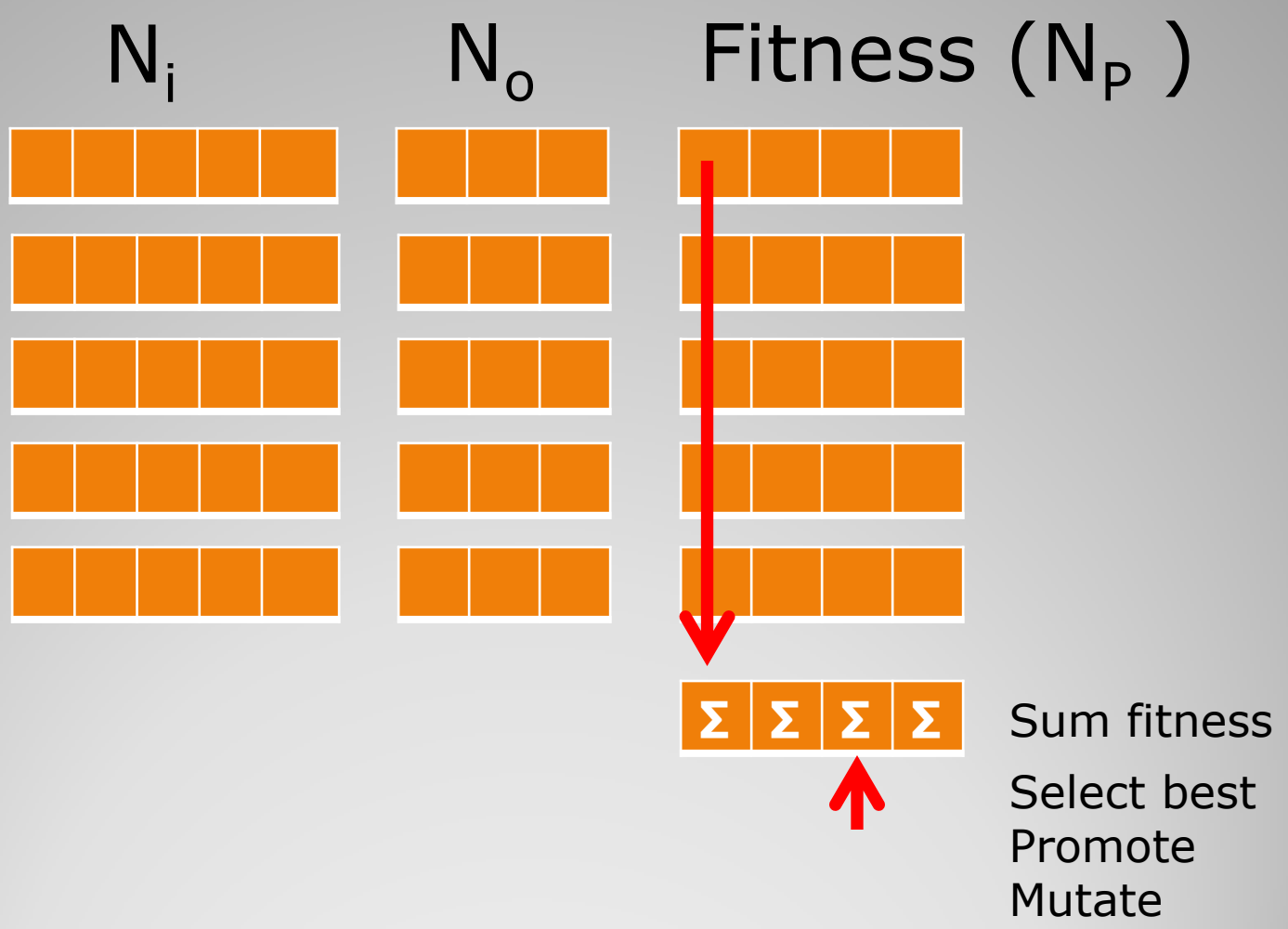


Test Cases

Testing one test case



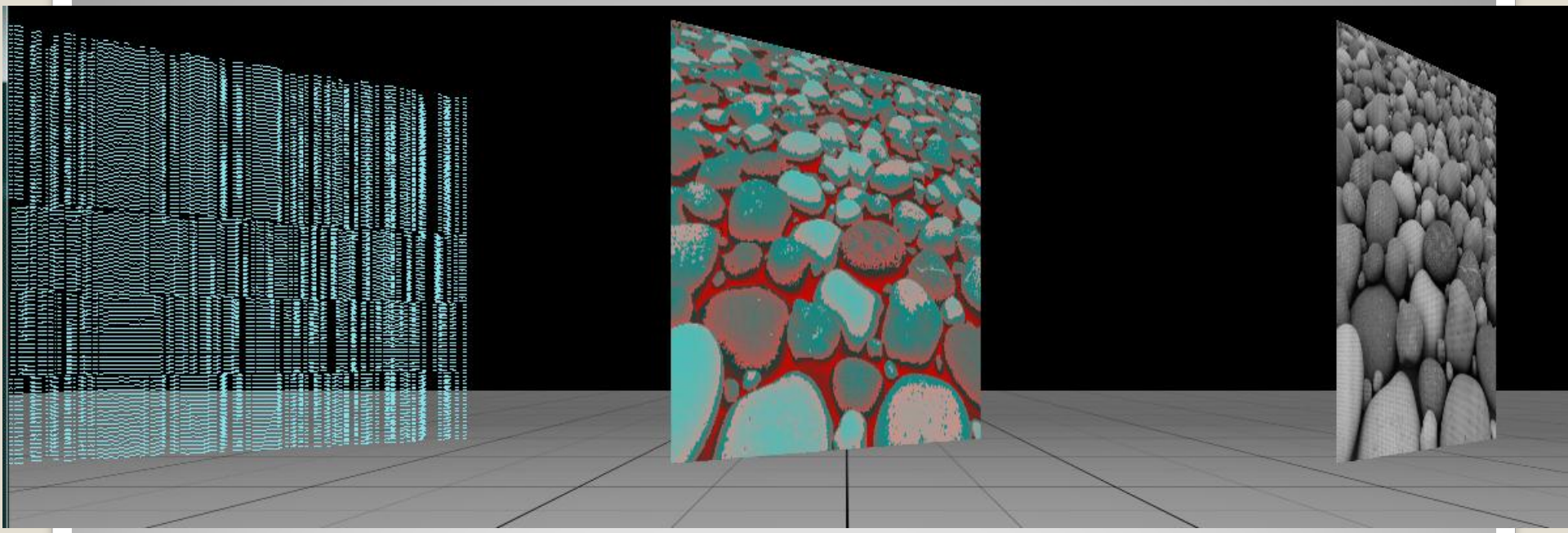
Typical vector operation:
Fitness = | actual – expected |^2



Finish the loop

- Test cases
- Population
- Mutator

Code



Test cases

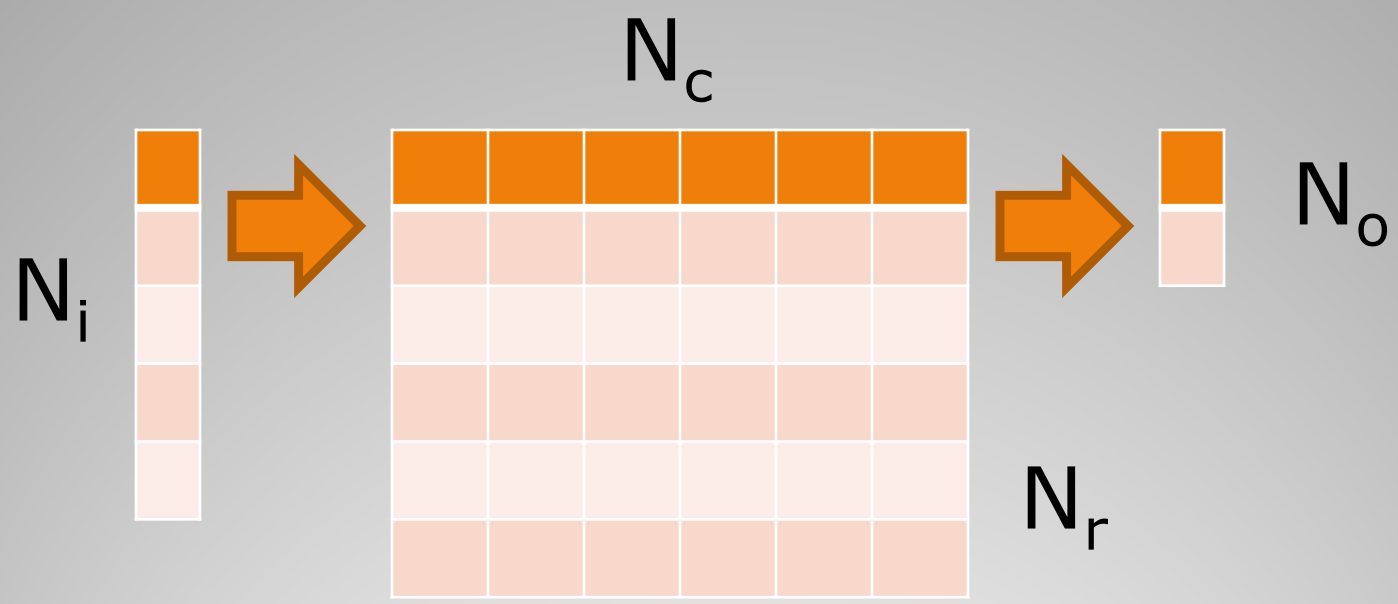
Validation Image

Bayer Image

Demo Explained

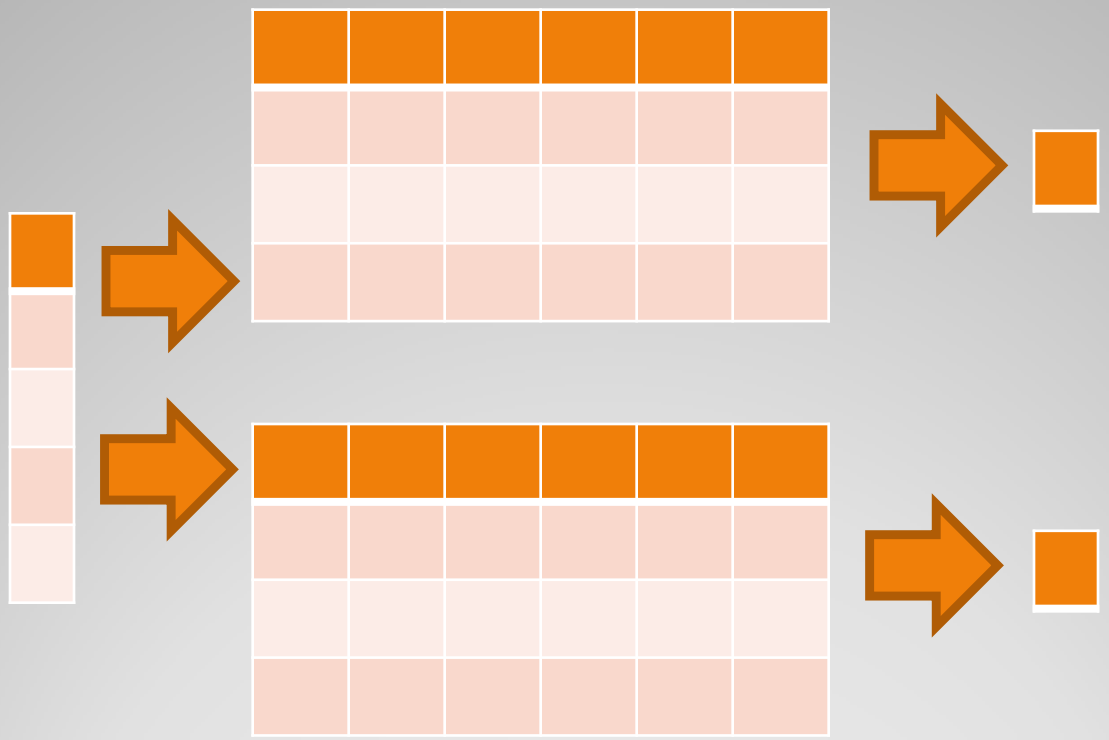
- CGP-based De-Bayer filter

Demo

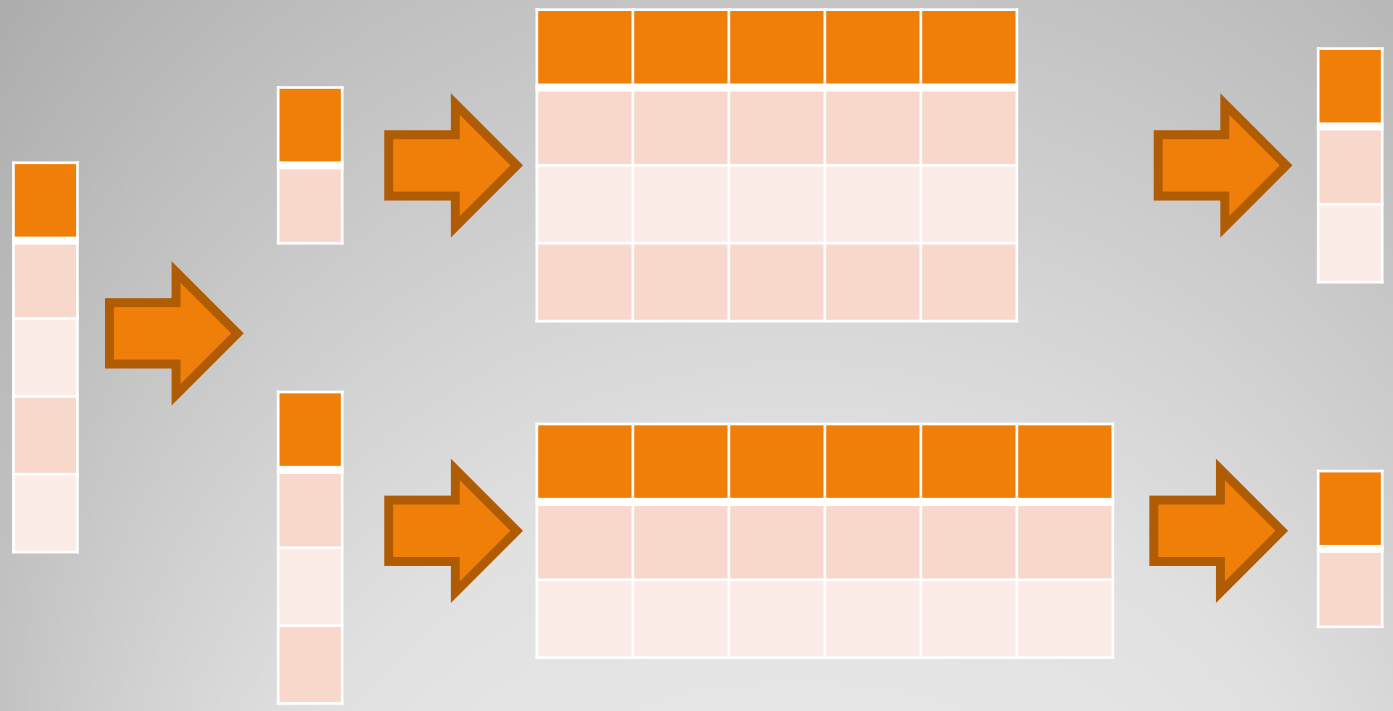


$$O(N_o^{2+})$$

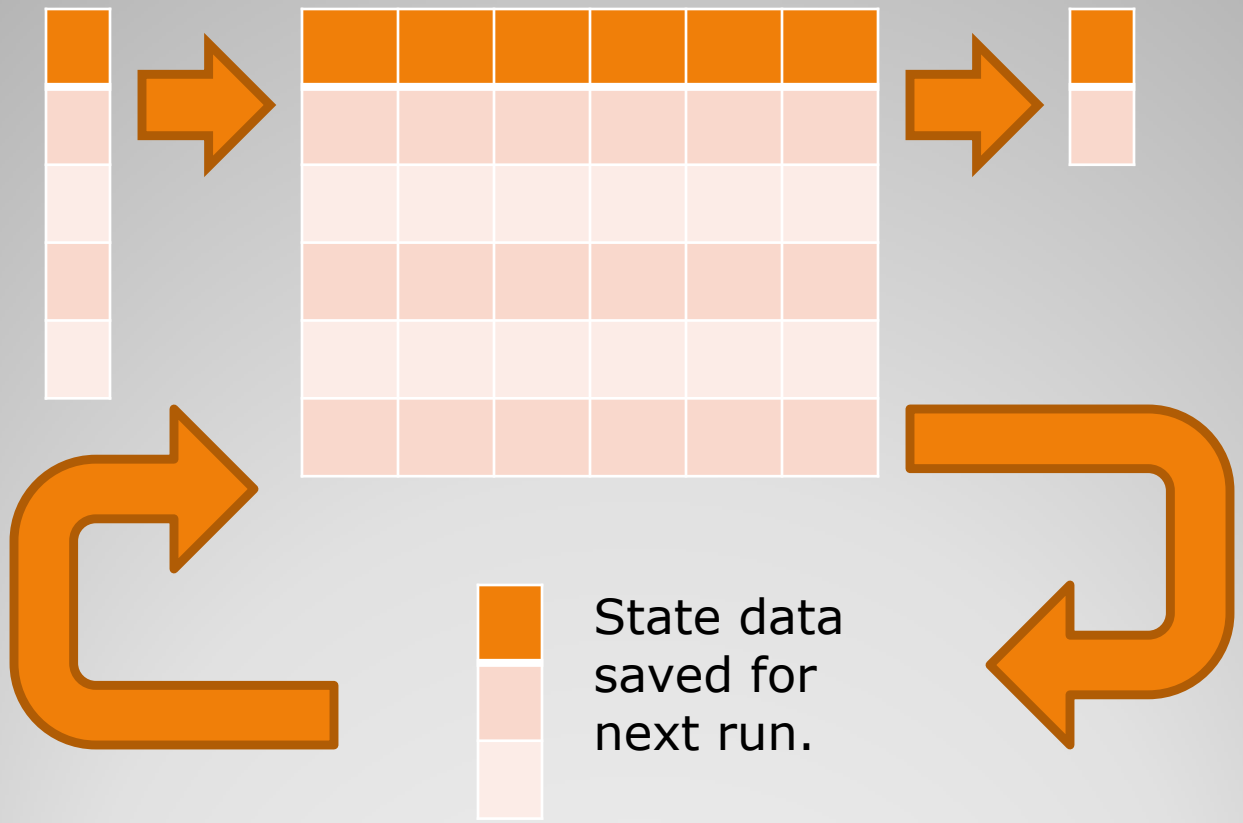
Scaling



Scaling – Simple Chromosomes



Scaling – Complex Chromosomes



Beyond functions - State

```
class Widget {
private:
        float fState[2];
        float gState[2];
        float sharedState[4];
public:
        void f(const float input[7],
               float output[3]);
        void g(const float input[4],
               float output[1]);
};
```

2 chromosomes with private and shared state

Tips

- Visualize!
- Tests should give partial success
- Increase population
- Keep mutation rate low enough

Future

- CGP on GPU
  - Especially validation!
- Islands
- More data visualizations
- Parameter tuning with PSO
- Open source

- Danke!
- <shameless recruiting ad>
  - Join us at ptscientists.com!
- </shameless recruiting ad>
- Wes Faler of Part-Time Scientists
  - wf@ptscientists.com

</presentation>

Resources

- Julian Miller (inventor of CGP)
  - http://sites.google.com/site/julianfrancismiller/professional
- CGP Book
  - "Cartesian Genetic Programming"
  - http://www.springer.com/computer/theoretical+computer+science/book/978-3-642-17309-7
- "Evolved to Win" e-book
  - http://www.moshesipper.com/etw/