# ECM3412/ECMM409
# Nature Inspired Computation
# Lecture 5

## Evolutionary Computation
### Encodings / Applications

# Encoding / Representation

**Maybe the main** issue in (applying) EC

Note that:

- Given an optimisation problem to solve, we need to find a way of encoding candidate solutions
- There can be many very different encodings for the same problem
- Each way affects the shape of the landscape and the choice of best strategy for climbing that landscape.

# Some Simple Direct Encodings

- Water Distribution Network Optimisation
  - There are *n* pipes with *m* possible diameters.
  - The problem is to create a new network design from these variables.
  - Each pipe must have a diameter, so the encoding is similar to a k-ary encoding: a chromosome of *n* integers long with 1->*m* possible values

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| D9 | D6 | D3 | D4 | D6 | D7 | D1 | D8 | D8 | D6 |
| p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 | p9 | p10 |

D = Diameter (pipe size and number of choices) p = pipe index

# Some Simple Direct Encodings

- Generalised Assignment Problem
  - If you have *n* workers and *m* jobs to complete, what encoding would you use?
  - Each worker can do more than one job, but each job takes a specified amount of the workers (finite) time.

# Encodings

- Encoding 1
  - Have an encoding of *n* integers, each with a range 1-*m*
    - \+ every worker has at least one job
    - \- not every job will have a worker assigned
    - \- cannot assign more than one job to a worker
    - \- one job may be assigned to 2 or more workers

| J9 | J6 | J3 | J4 | J6 | J7 | J1 | J8 | J8 | J6 |
|---|---|---|---|---|---|---|---|---|---|
| w1 | w2 | w3 | w4 | w5 | w6 | w7 | w8 | w9 | w10 |

- Encoding 2
  - Have an encoding of *m* integers, each with a range 1-*n*
    - \+ every job has a worker assigned to it
    - \- a worker may be overworked

| W9 | W6 | W3 | W4 | W6 | W7 | W1 | W8 | W8 | W6 |
|---|---|---|---|---|---|---|---|---|---|
| j1 | j2 | j3 | j4 | j5 | j6 | j7 | j8 | j9 | j10 |

# E.g. encoding a timetable I

4, 5, 13, 1, 1, 7, 13, 2

Exam2 in 5th slot

Exam1 in 4th slot

Etc …

| | mon | tue | wed | thur |
|---|---|---|---|---|
| 9:00 | E4, E5 | E2 | | E3, E7 |
| 11:00 | E8 | | | |
| 2:00 | | E6 | | |
| 4:00 | E1 | | | |

- Generate *any* string of *8* numbers between 1 and 16, and we have a timetable!
- Fitness may be <clashes> + <consecs> + etc …
- Figure out an encoding, and a fitness function, and you can try to evolve solutions.

# Mutating a Timetable with Encoding 1

4, 5, 13, 1, 1, 7, 13, 2

| | mon | tue | wed | thur |
|---|---|---|---|---|
| 9:00 | E4, E5 | E2 | | E3, E7 |
| 11:00 | E8 | | | |
| 2:00 | | E6 | | |
| 4:00 | E1 | | | |

Using straightforward single-gene mutation

Choose a random gene

# Mutating a Timetable with Encoding 1

4, 5, 6 , 1, 1, 7, 13, 2

| | mon | tue | wed | thur |
|---|---|---|---|---|
| 9:00 | E4, E5 | E2 | | E7 |
| 11:00 | E8 | E3 | | |
| 2:00 | | E6 | | |
| 4:00 | E1 | | | |

Using straightforward single-gene mutation

One mutation changes position of one exam

# Encoding a timetable II

| | mon | tue | wed | thur |
|---|---|---|---|---|
| 9:00 | E4 E5 | E2 | | |
| 11:00 | | | E3 | |
| 2:00 | E8 | | | E7 |
| 4:00 | E1 | E6 | | |

4, 5, 10, 1, 1, 7, 15, 2

Etc …

Use the 10th clash-free slot for exam3

Use the 5th clash-free slot for exam2

Use the 4th clash-free slot for exam1

Suppose these groups would clash (e.g. non-consecutive, non-parallel)

{E1, E2}, {E1,E3}, {E2, E6}, {E2,E7}, {E2,E8}, {E3,E5}, {E3,E6}, {E4,E6}, {E4, E7}, {E5, E7}, {E5, E8}, {E6, E8}

# Mutation with Encoding II

4, 5, 10, 1, 1, 7, 15, 2

| | mon | tue | wed | thur |
|---|---|---|---|---|
| 9:00 | E4 E5 | E2 | | |
| 11:00 | | | E3 | |
| 2:00 | E8 | | | E7 |
| 4:00 | E1 | E6 | | |

Use the 10th clash-free slot for exam3

Use the 5th clash-free slot for exam2

Use the 4th clash-free slot for exam1

Suppose these groups would clash

{E1, E2}, {E1,E3}, {E2, E6}, {E2,E7}, {E2,E8}, {E3,E5}, {E3,E6}, {E4,E6}, {E4, E7}, {E5, E7}, {E5, E8}, {E6, E8}

# Mutation with Encoding II

4, 5, 10, 1, 14, 7, 15, 2

| | mon | tue | wed | thur |
|---|---|---|---|---|
| 9:00 | E4 | E2 | | |
| 11:00 | E8 | | E3 | E5 |
| 2:00 | | | | |
| 4:00 | E1 | E6 | | E7 |

Use the 13th clash-free slot for exam3

Use the 5th clash-free slot for exam2

Use the 4th clash-free slot for exam1

Suppose these groups would clash

{E1, E2}, {E1,E3}, {E2, E6}, {E2,E7}, {E2,E8}, {E3,E5}, {E3,E6}, {E4,E6}, {E4, E7}, {E5, E7}, {E5, E8}, {E6, E8}

# Direct vs Indirect Representation

| Representation | Encoding | Modifies | Invalid Solutions |
|---|---|---|---|
| Direct | 1,2,3,4<br>011010<br>ABFCDE | A variable in the problem<br>E.g. Exam Time, Water Pipe Size | Dealt with solely by penalising fitness |
| Indirect | 1,2,3,4<br>011010<br>ABFCDE | The variables of a heuristic<br>E.g. Clash-Free Exam Time, Rules for Sizing Water Pipes | Mostly dealt with by encoding. Some penalising by fitness if necessary |

# Direct vs Indirect Encodings

Direct:

- straightforward genotype (encoding) → phenotype (individual) mapping
- Easy to estimate effects of mutation
- Fast interpretation of chromosome (hence speedier fitness evaluation)

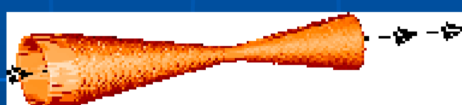
Indirect:

- Easier to exploit domain knowledge
- Hence, possible to `encode away' undesirable features
- Hence, can seriously cut down the size of the search space
- But, slow interpretation
- Neighbourhoods are highly rugged.

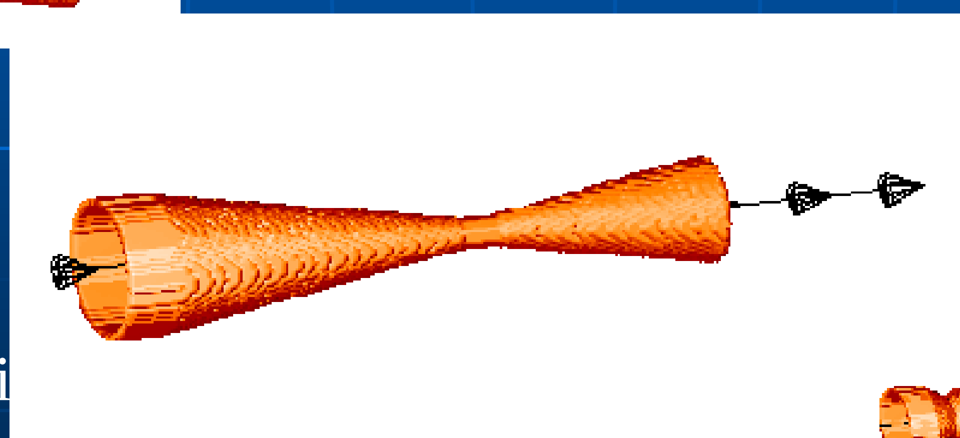
One of the very first applications. Determine the internal shape of a two-phase jet nozzle that can achieve the maximum possible thrust under given starting conditions

Ingo Rechenberg was the *very* first, with pipe-bend design

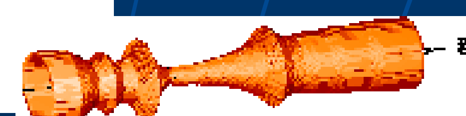
This is slightly later work in the same lab, by Schwefel



Starting point

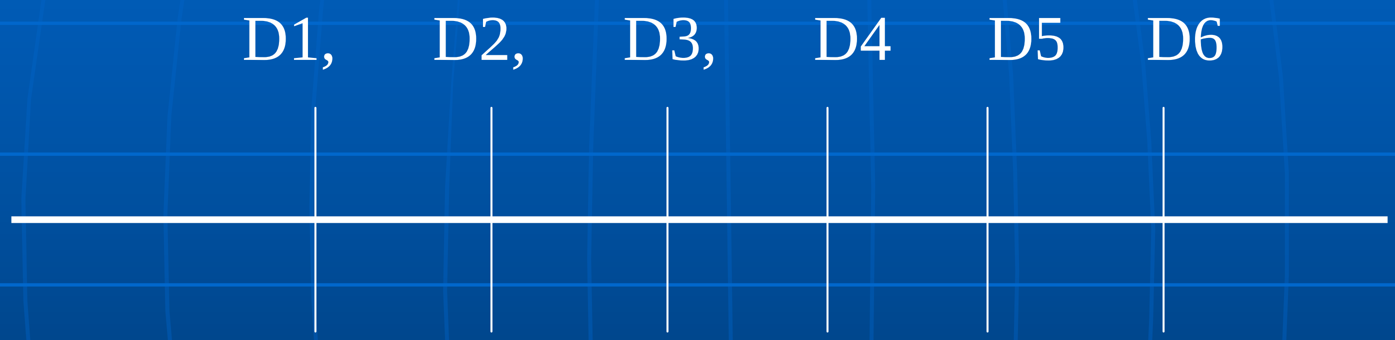


EA (ES) runni



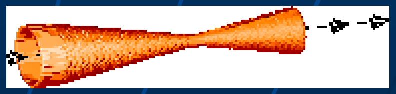
A recurring theme: design freedom → entirely new and better designs based on principles we don't yet understand.

# A Real Encoding (and: How EAs can innovate, rather than just optimize)



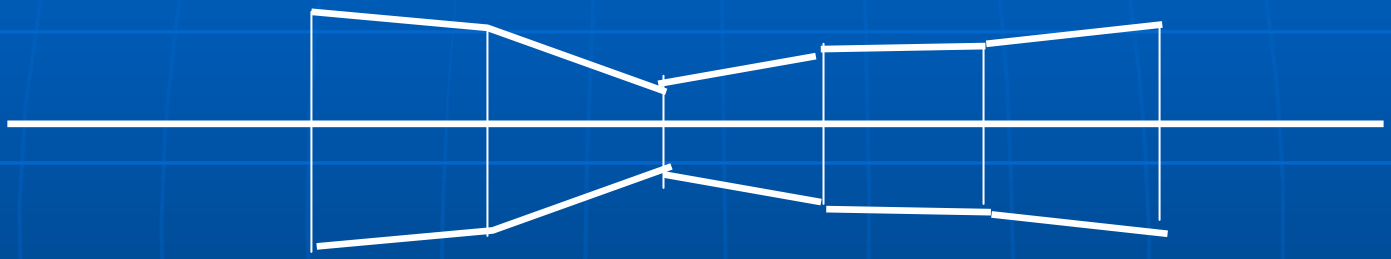

D1 >= D2 >= D3, D4 <= D5 <= D6

Fixed at six diameters, five sections



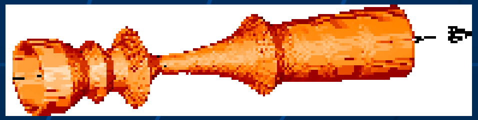
# An example genotype and its phenotype

2, 1.8, 1.1, 1.3 1.3 1.5



Assume each is allowed to vary between 0.1 and 2.

What is the likely effect on fitness of *random-gene* mutation where we replace a gene with a random new value in the range?



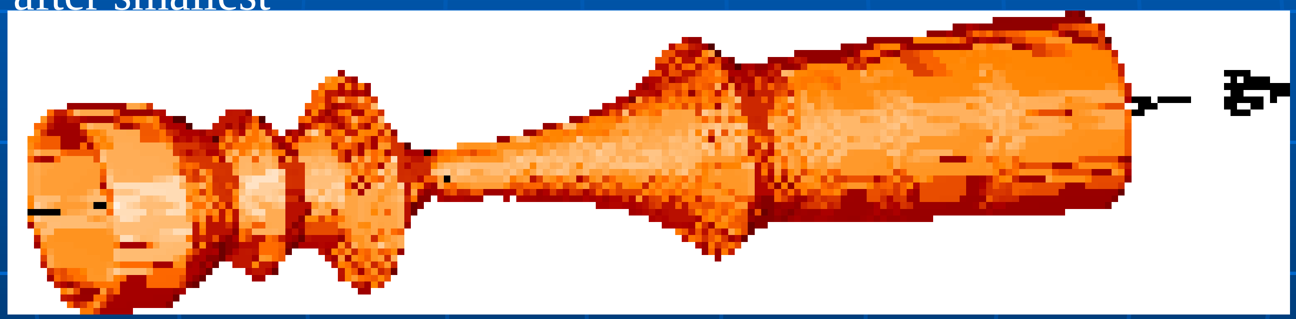
# E.g. How EAs can innovate, rather than just optimize

Num sections before smallest

Section diameters

Z1, Z2, D1, D2, D3 Dsmall…, Dn, Dn+1, …

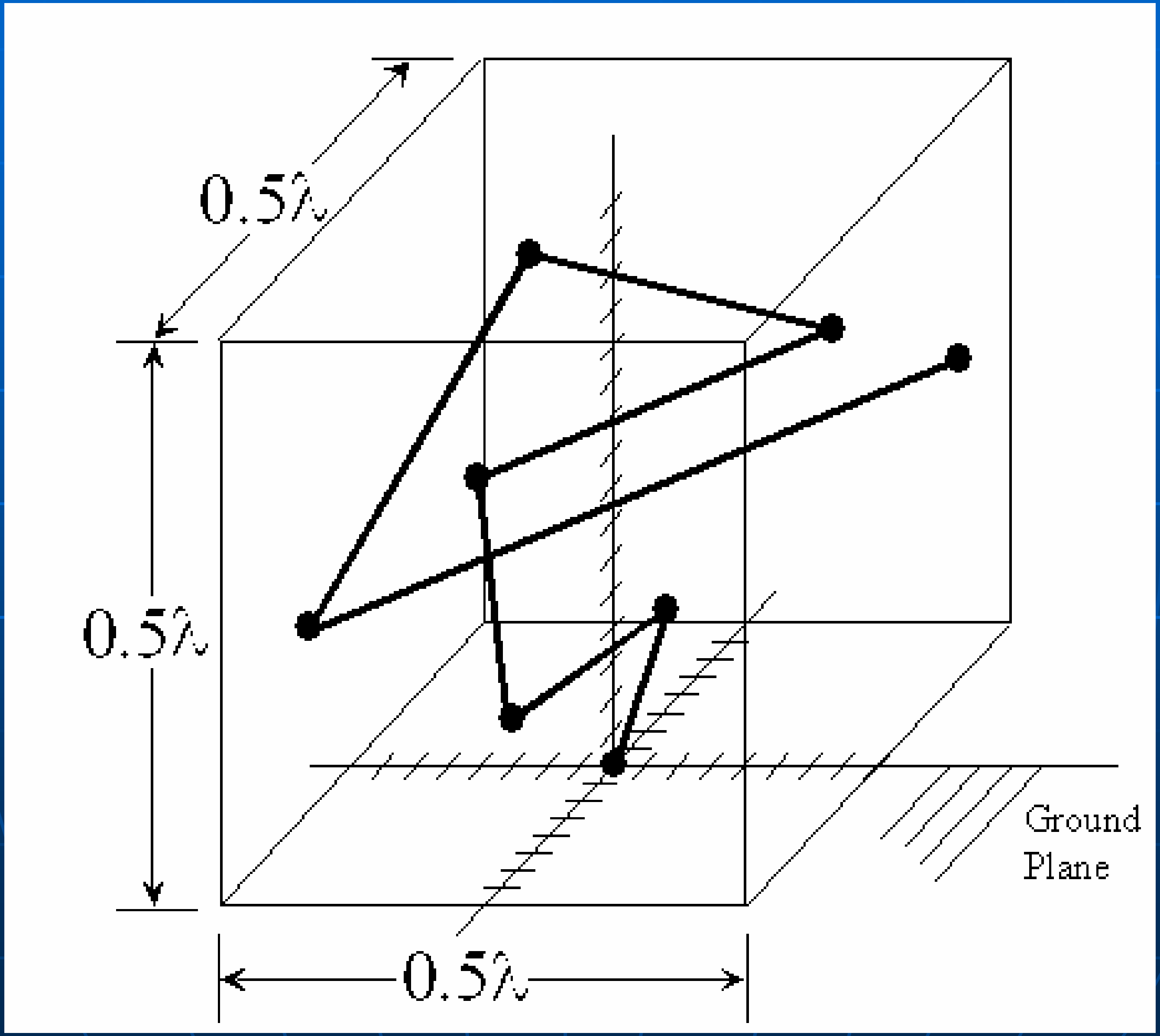
Num sections after smallest



Middle section constrained to be smallest,
That’s all
Mutations can change diameters, add sections, and delete sections

# Some slides from an Introductory lecture by John Koza, Stanford

# ANTENNA DESIGN



# ANTENNA DESIGN

- **The problem (Altshuler and Linden 1998) is to determine the *x-y-z* coordinates of the 3-dimensional position of the ends (*X1*, *Y1*, *Z1*, *X2*, *Y2*, *Z2*,... , *X7*, *Y7*, *Z7*) of 7 straight wires so that the resulting 7-wire antenna satisfies certain performance requirements**
- **The first wire starts at feed point (0, 0, 0) in the middle of the ground plane**
- **The antenna must fit inside the 0.5λ cube**

# ANTENNA GENOME

| $X_1$ | $Y_1$ | $Z_1$ | $X_2$ | $Y_2$ | $Z_2$ | … |
|---|---|---|---|---|---|---|
| +0010 | -1110 | +0001 | +0011 | -1011 | +0011 | … |

- **105-bit chromosome (genome)**
- **Each *x-y-z* coordinate is represented by 5 bits (4-bit granularity for data plus a sign bit)**
- **Total chromosome is 3 × 7 × 5 = 105 bits**

*This was done with binary encoding – the fashion at the time, and still often used, but real-valued or integer coding will be just as applicable*

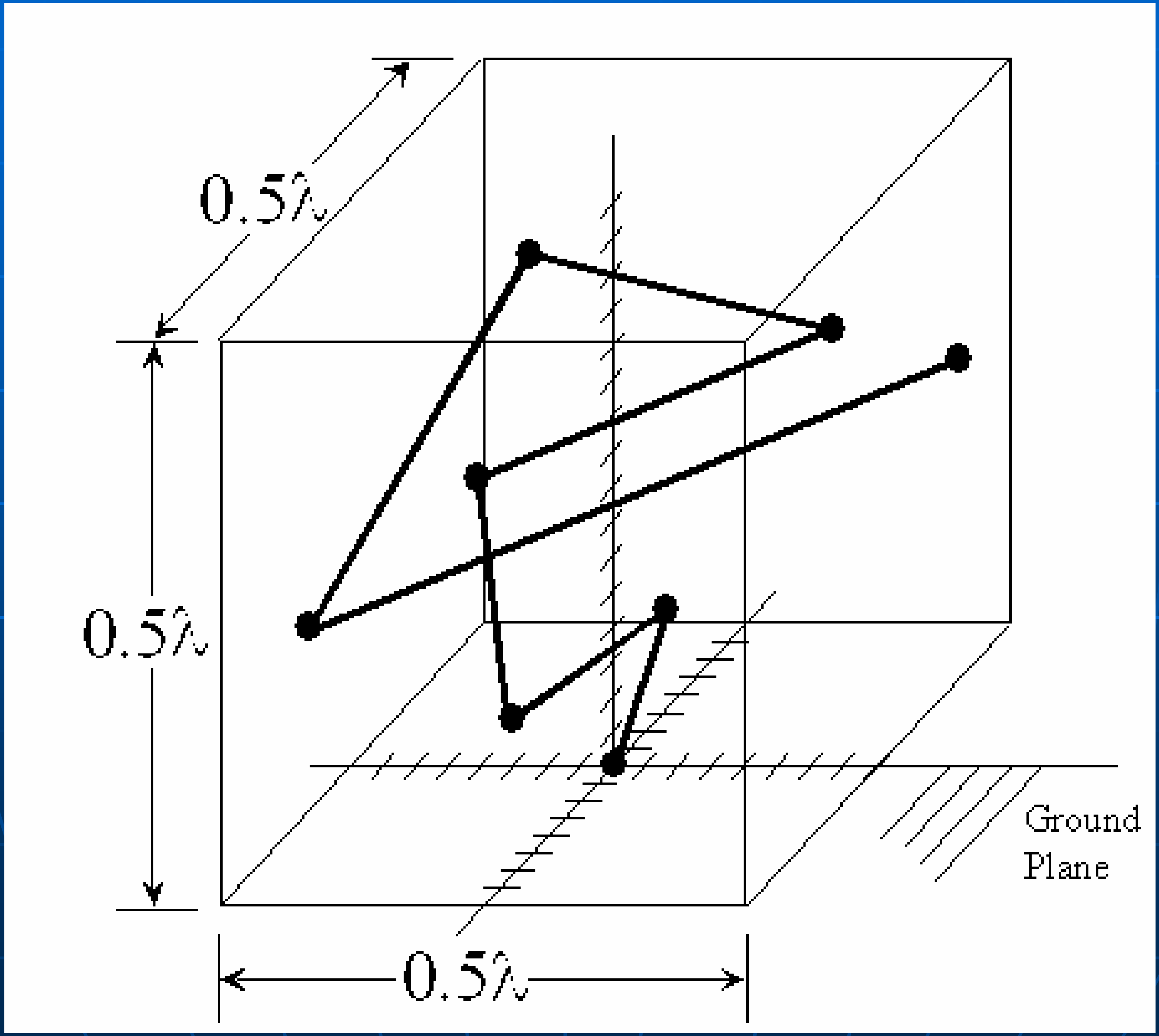
# ANTENNA FITNESS

- Antenna is for ground-to-satellite communications for cars and handsets
- We desire near-uniform gain pattern 10° above the horizon
- Fitness is measured based on the antenna's radiation pattern. The radiation pattern is simulated by National Electromagnetics Code (NEC)

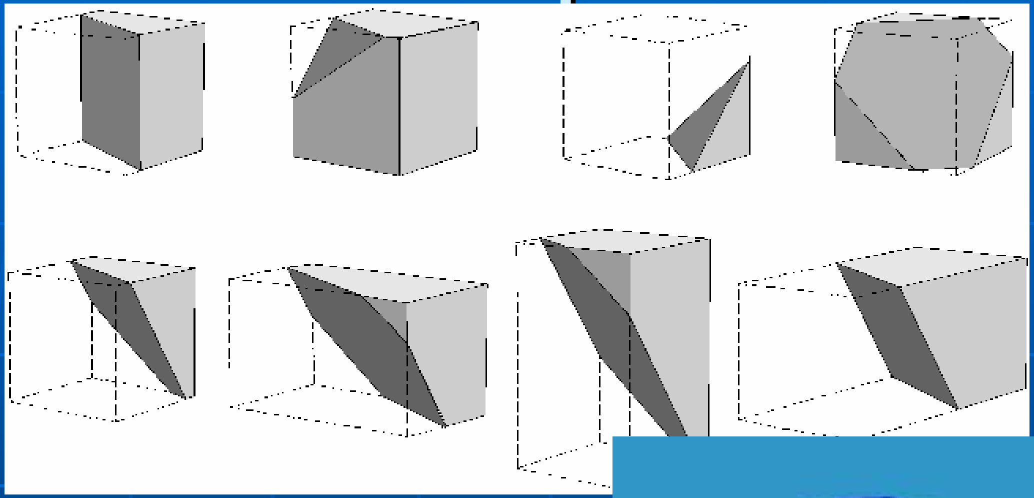
# ANTENNA FITNESS

- Fitness is sum of the squares of the difference between the average gain and the antenna's gain
- Sum is taken for angles Θ between -90° and +90° and all azimuth angles Φ from 0° to 180°
- The smaller the value of fitness, the better

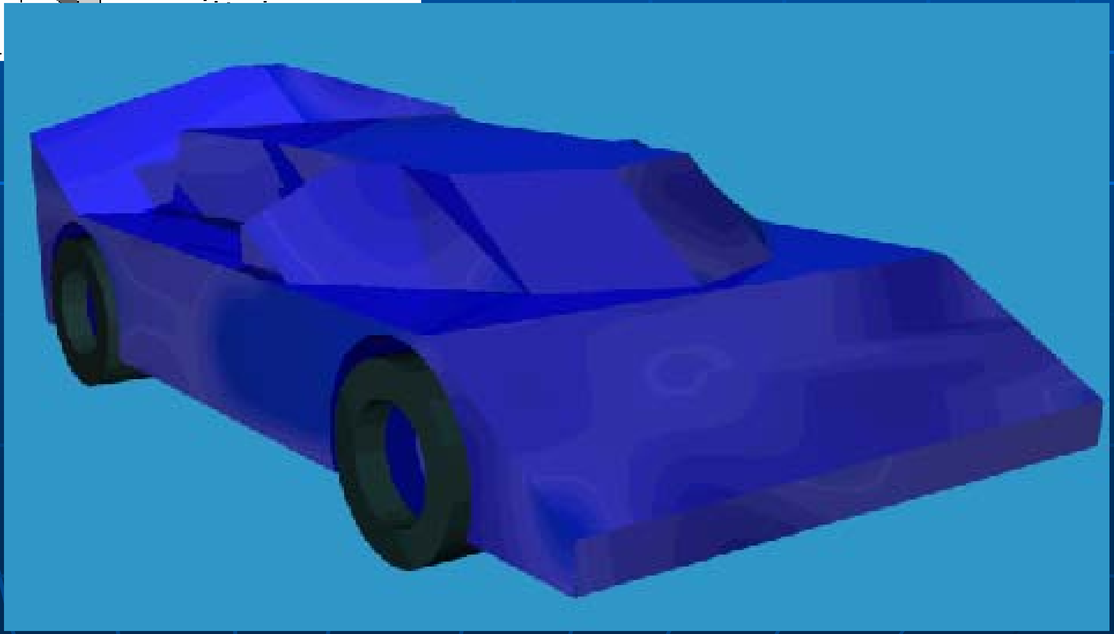
# U. S. PATENT 5,719,794



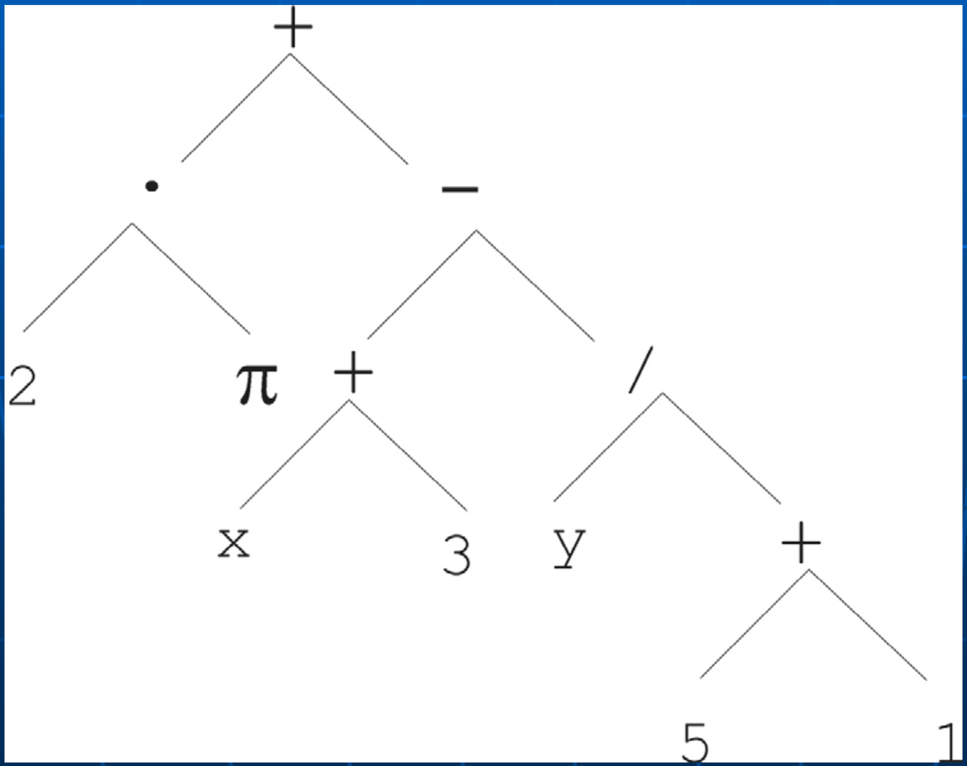
# Component-based representations



Bentley's work used primitive shapes to construct novel designs



# Tree based representation




$$2 \cdot \pi + \left( (x+3) - \frac{y}{5+1} \right)$$

# Next time: Genetic Programming