ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN

VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

한-베정보통신기술대학교

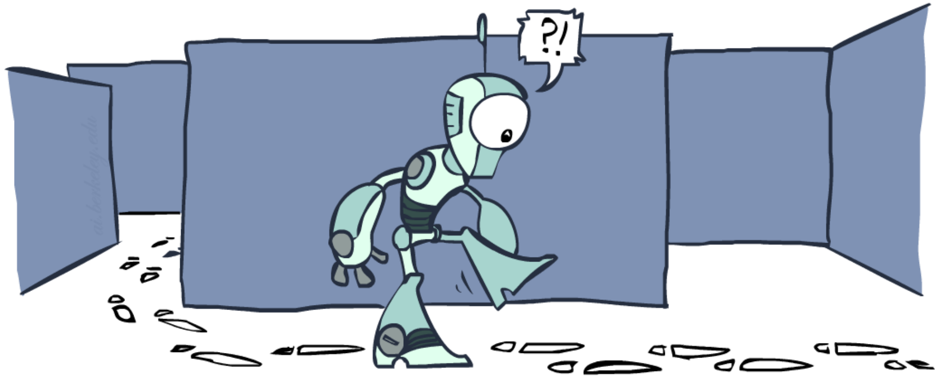
Nhân bản – Phụng sự – Khai phóng

Evolutionary Computation

Artificial Intelligence

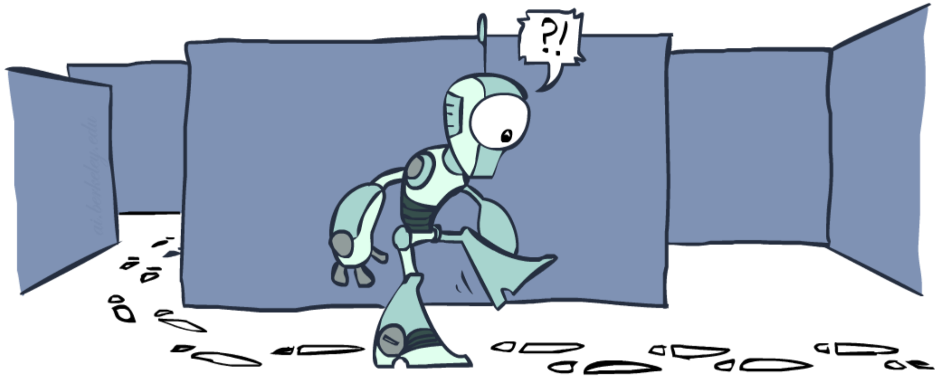
CONTENTS

- Recap of EC metaphor
- Evolutionary Algorithm
- Genetic Algorithm



CONTENTS

- **Recap of EC metaphor**
- Evolutionary Algorithm
- Genetic Algorithm



Recap of EC metaphor (1/2)

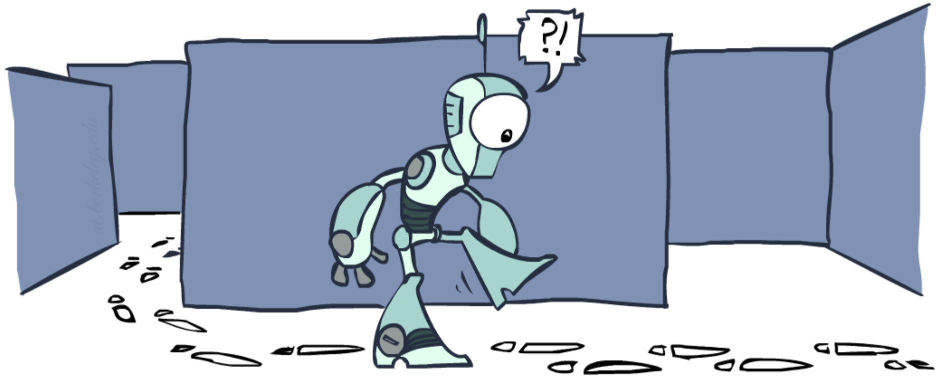
- A population of individuals exists in an environment with limited resources
- ***Competition*** for those resources causes selection of those ***fitter*** individuals that are better adapted to the environment
- These individuals act as seeds for the generation of new individuals through recombination and mutation
- The new individuals have their fitness evaluated and compete (possibly also with parents) for survival.
- Over time ***Natural selection*** causes a rise in the fitness of the population

Recap of EC metaphor (2/2)

- EAs fall into the category of “generate and test” algorithms
- They are stochastic, population-based algorithms
- Variation operators (recombination and mutation) create the necessary diversity and thereby facilitate novelty
- Selection reduces diversity and acts as a force pushing quality

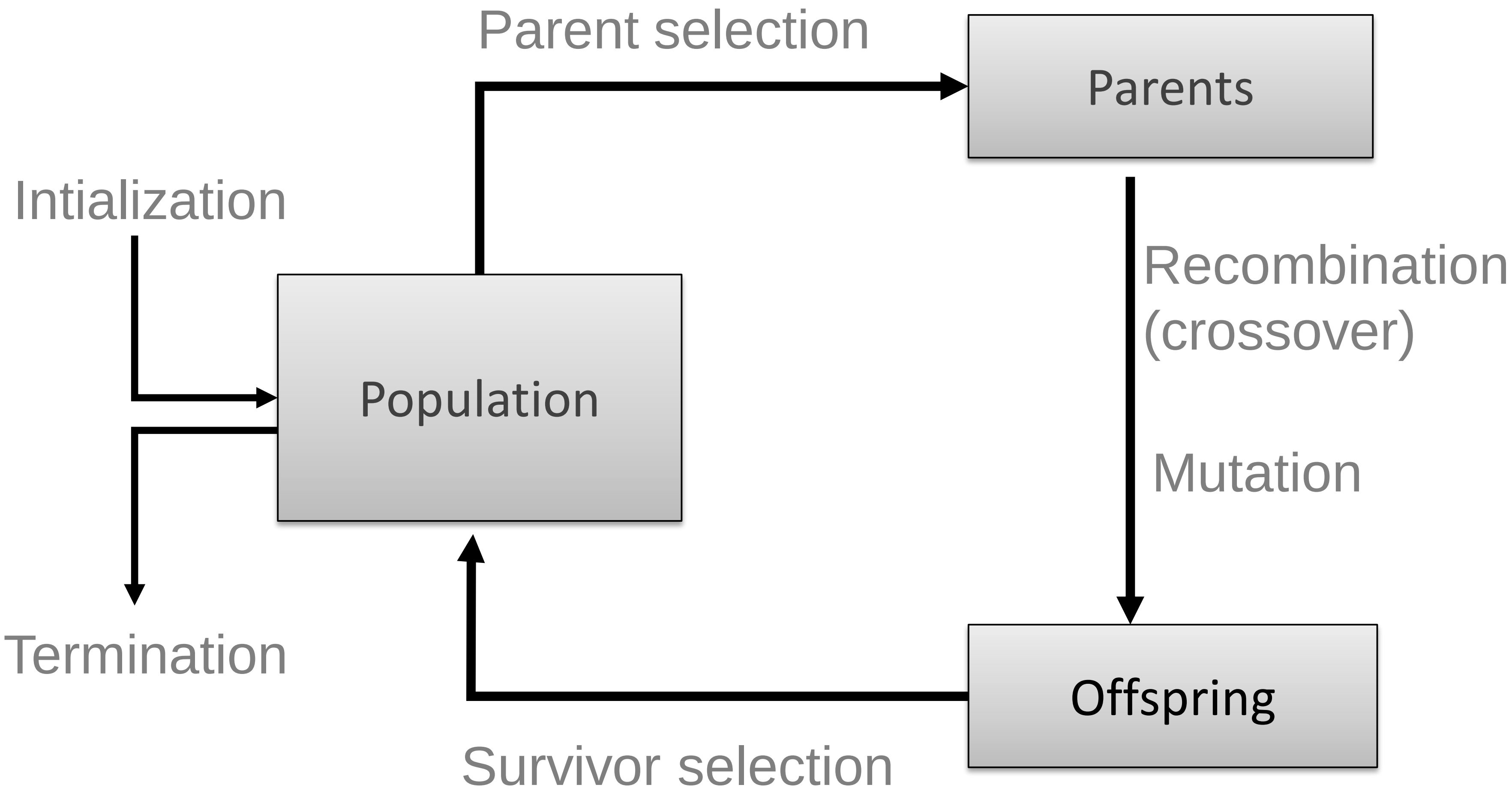
CONTENTS

- Recap of EC metaphor
- **Evolutionary Algorithm**
- Genetic Algorithm



- **Scheme of an EA**
- **Main EA components:**
  - Representation / evaluation / population
  - Parent selection / survivor selection
  - Recombination / mutation
- **Examples: eight-queens problem**
- **Typical EA behaviour**
- **EAs and global optimisation**
- **EC and neighbourhood search**

Scheme of an EA: General scheme of EAs



Scheme of an EA: EA scheme in pseudo-code

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Scheme of an EA: Common model of evolutionary processes

- Population of individuals
- Individuals have a fitness
- Variation operators: crossover, mutation
- Selection towards higher fitness
  - “survival of the fittest” and
  - “mating of the fittest”

Neo Darwinism:

Evolutionary progress towards higher life forms

=

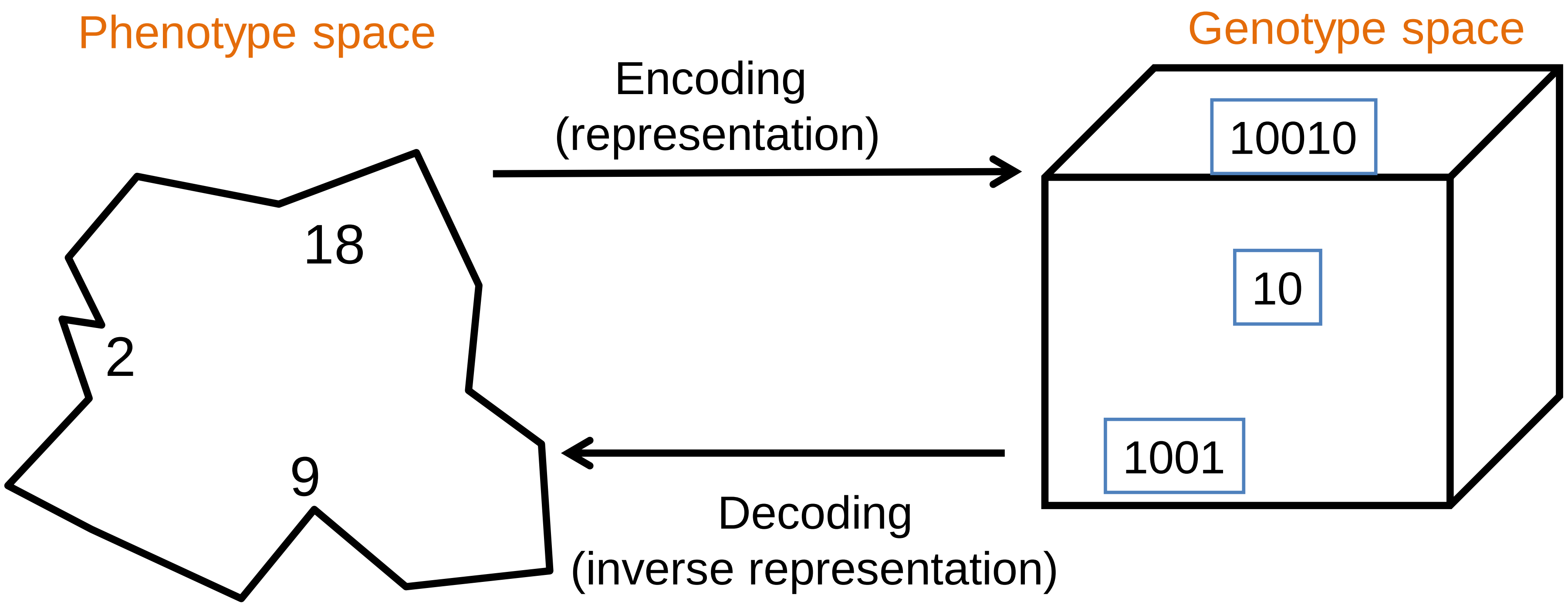
Optimization according to some fitness-criterion
(optimization on a fitness landscape)

Main EA components: Representation (1/2)

- **Role: provides code for candidate solutions that can be manipulated by variation operators**
- **Leads to two levels of existence**
  - phenotype: object in original problem context, the outside
  - genotype: code to denote that object, the inside (chromosome, "digital DNA")
- **Implies two mappings:**
  - Encoding : phenotype=> genotype (not necessarily one to one)
  - Decoding : genotype=> phenotype (must be one to one)
- **Chromosomes contain genes, which are in (usually fixed) positions called loci (sing. locus) and have a value (allele)**

Main EA components: Representation (2/2)

Example: represent integer values by their binary code



In order to find the global optimum, every feasible solution must be represented in genotype space

Main EA components: Evaluation (fitness) function

- **Role:**
  - Represents the task to solve, the requirements to adapt to (can be seen as "the environment")
  - Enables selection (provides basis for comparison)
  - e.g., some phenotypic traits are advantageous, desirable, e.g. big ears cool better, these traits are rewarded by more offspring that will expectedly carry the same trait
- **A.k.a. *quality* function or *objective* function**
- **Assigns a single real-valued fitness to each phenotype which forms the basis for selection**
  - So the more discrimination (different values) the better
- **Typically we talk about fitness being maximised**
  - Some problems may be best posed as minimisation problems, but conversion is trivial

Main EA components: Population (1/2)

- Role: holds the candidate solutions of the problem as individuals (genotypes)
- Formally, a population is a multiset of individuals, i.e. repetitions are possible
- Population is the basic unit of evolution, i.e., the population is evolving, not the individuals
- Selection operators act on population level
- Variation operators act on individual level

Main EA components: Population (2/2)

- Some sophisticated EAs also assert a spatial structure on the population e.g., a grid
- Selection operators usually take whole population into account i.e., reproductive probabilities are *relative* to *current* generation
- **Diversity** of a population refers to the number of different fitnesses / phenotypes / genotypes present (note: not the same thing)

Role:

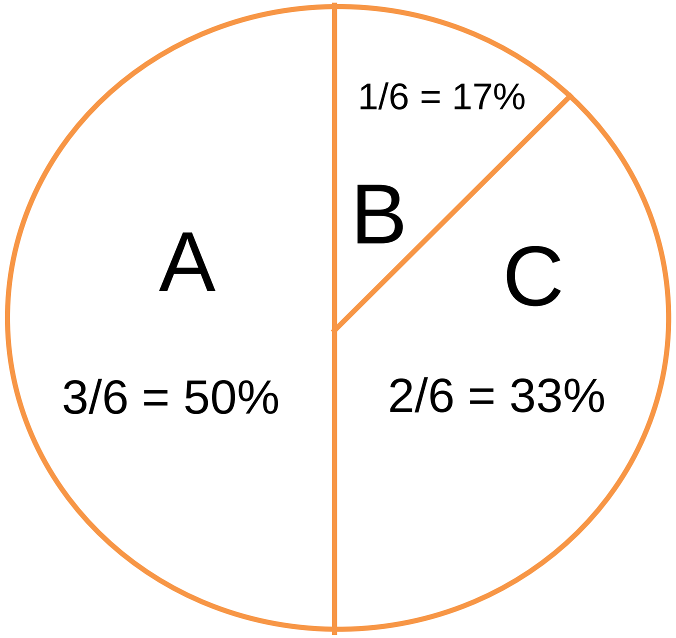
- **Identifies individuals**
  - to become parents
  - to survive
- **Pushes population towards higher fitness**
- **Usually probabilistic**
  - high quality solutions more likely to be selected than low quality
  - but not guaranteed
  - even worst in current population usually has non-zero probability of being selected
- **This *stochastic* nature can aid escape from local optima**

Example: roulette wheel selection

fitness(A) = 3

fitness(B) = 1

fitness(C) = 2



In principle, any selection mechanism can be used for parent selection as well as for survivor selection

- **Survivor selection A.k.a. *replacement***
- **Most EAs use fixed population size so need a way of going from (parents + offspring) to next generation**
- **Often deterministic (while parent selection is usually stochastic)**
  - Fitness based : e.g., rank parents + offspring and take best
  - Age based: make as many offspring as parents and delete all parents
- **Sometimes a combination of stochastic and deterministic (elitism)**

Main EA components: Variation operators

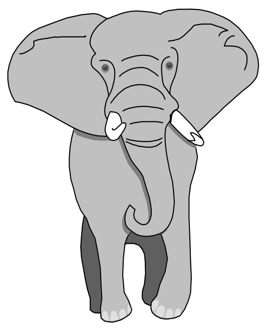
- **Role: to generate new candidate solutions**
- **Usually divided into two types according to their arity (number of inputs):**
  - Arity 1 : mutation operators
  - Arity >1 : recombination operators
  - Arity = 2 typically called crossover
  - Arity > 2 is formally possible, seldom used in EC
- **There has been much debate about relative importance of recombination and mutation**
  - Nowadays most EAs use both
  - Variation operators must match the given representation

Main EA components: Mutation (1/2)

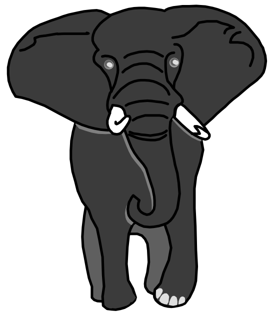
- **Role: causes small, random variance**
- **Acts on one genotype and delivers another**
- **Element of randomness is essential and differentiates it from other unary heuristic operators**
- **Importance ascribed depends on representation and historical dialect:**
  - Binary GAs – background operator responsible for preserving and introducing diversity
  - EP for FSM's / continuous variables – only search operator
  - GP – hardly used
- **May guarantee connectedness of search space and hence convergence proofs**

Main EA components: Mutation (2/2)

before 1 1 1 1 1 1 1 →



after 1 1 1 0 1 1 1 →

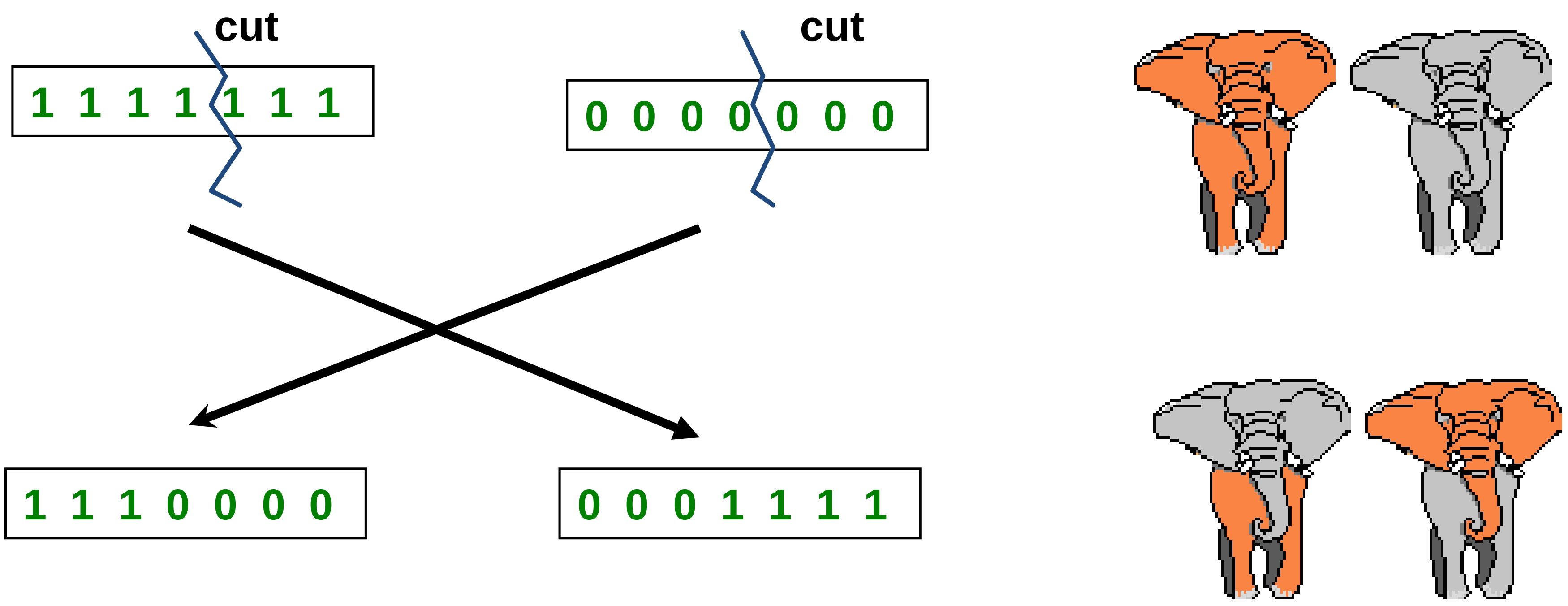


Main EA components: Recombination (1/2)

- Role: merges information from parents into offspring
- Choice of what information to merge is stochastic
- Most offspring may be worse, or the same as the parents
- Hope is that some are better by combining elements of genotypes that lead to good traits
- Principle has been used for millennia by breeders of plants and livestock

Main EA components: Recombination (2/2)

Parents



Offspring

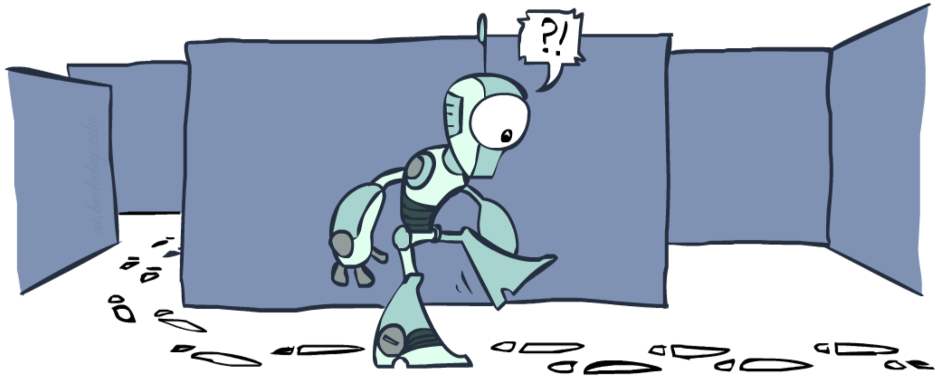
Main EA components: Initialisation / Termination

- **Initialisation usually done at random,**
  - Need to ensure even spread and mixture of possible allele values
  - Can include existing solutions, or use problem-specific heuristics, to “seed” the population

- **Termination condition checked every generation**
  - Reaching some (known/hoped for) fitness
  - Reaching some maximum allowed number of generations
  - Reaching some minimum level of diversity
  - Reaching some specified number of generations without fitness improvement

CONTENTS

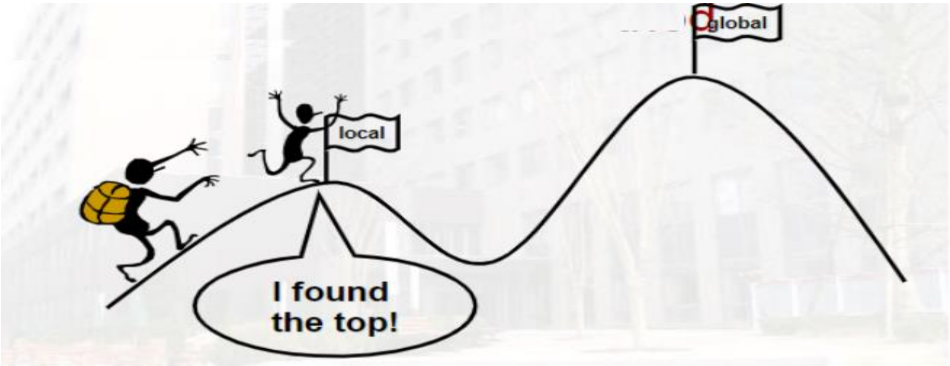
- Recap of EC metaphor
- Evolutionary Algorithm
- **Genetic Algorithm**



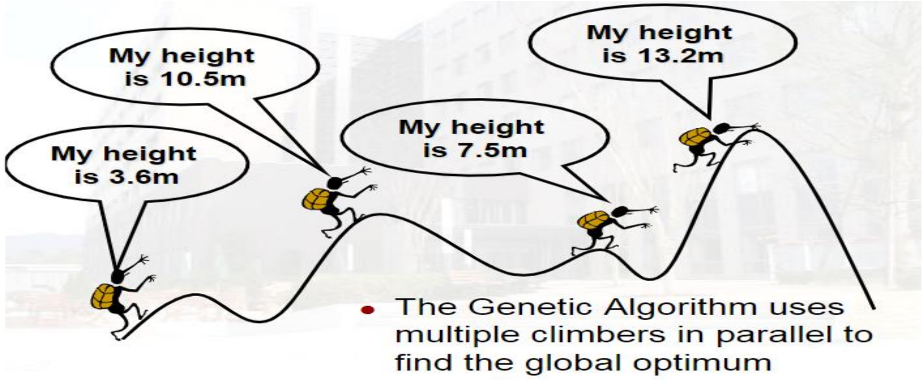
Genetic algorithm

- **John Holland in 1975**
- **A subset of Evolutionary Computation**
- **A search-based optimization technique based on the principles of Genetics and Natural Selection**
  - to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve
  - to solve optimization problems, in research, and in machine learning
- **Very popular in various research community**

- Genetic Algorithms have the ability to deliver a “good-enough” solution “fast-enough”.



- The reasons why GAs are needed:
  - Solving Difficult Problems
  - Failure of Gradient Based Methods
  - Getting a Good Solution Fast



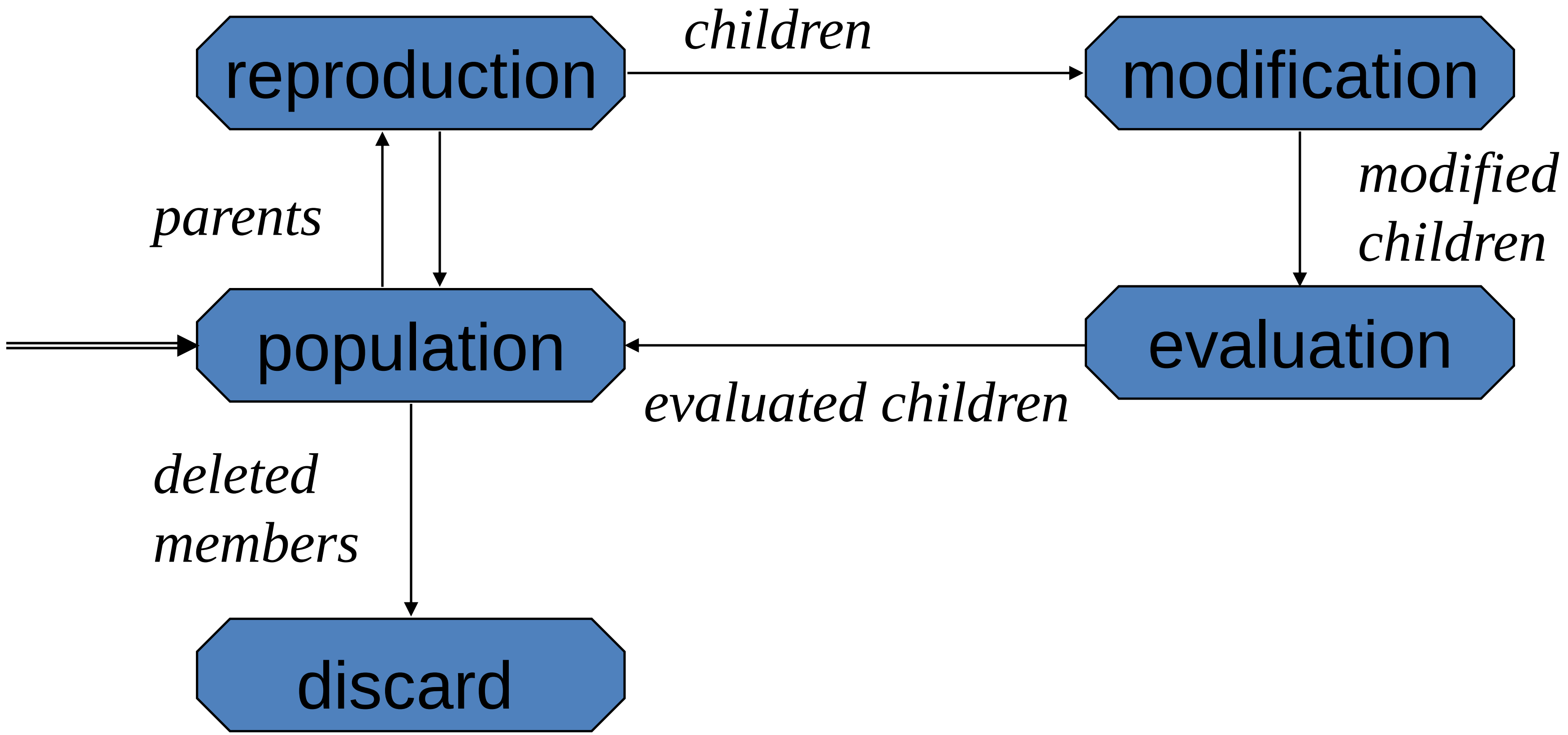
Components of a GA

A problem to solve, and ...

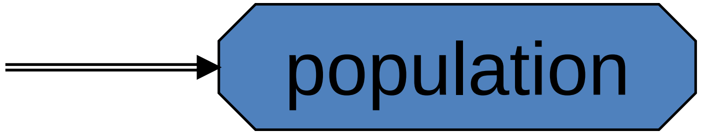
- Encoding technique (*chromosome*)
- Initialization procedure (*creation)*
- Evaluation function *(environment)*
- Selection of parents (*reproduction)*
- Genetic operators *(mutation, crossover)*

```
{
    initialize population;
    evaluate population;
    while TerminationCriteriaNotSatisfied
    {
        select parents for reproduction;
        perform recombination and mutation;
        evaluate population;
    }
}
```

The GA Cycle of Reproduction

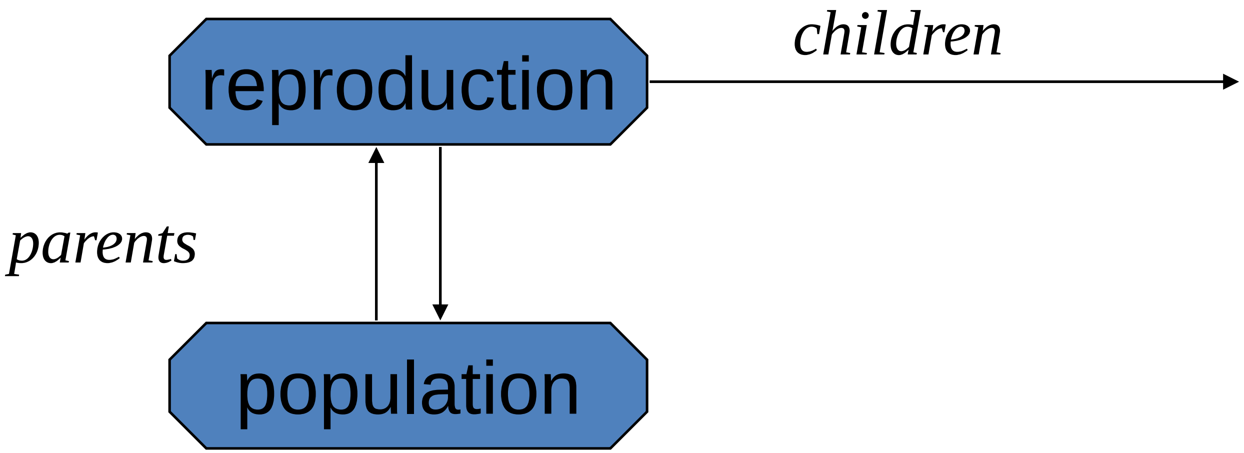


Population



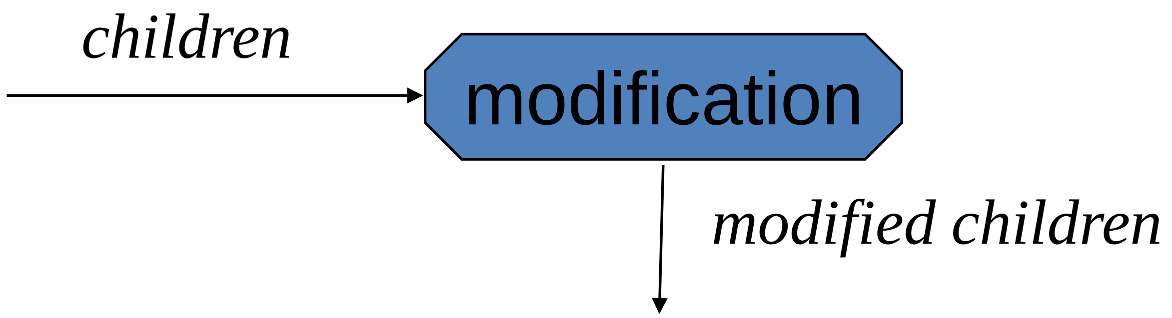
Chromosomes could be:

- Bit strings (0101 ... 1100)
- Real numbers (43.2 -33.1 ... 0.0 89.2)
- Permutations of element (E11 E3 E7 ... E1 E15)
- Lists of rules (R1 R2 R3 ... R22 R23)
- ... any data structure ...



Parents are selected at random with selection chances biased in relation to chromosome evaluations.

Chromosome Modification



- **Modifications are stochastically triggered**
- **Operator types are:**
  - Mutation
  - Crossover (recombination)

Mutation: Local Modification

Before:	(1 0 1 $\boxed{1}$ 0 1 1 0)
After:	(0 1 1 $\boxed{0}$ 0 1 1 0)

Before:	(1.38 $\boxed{-69.4}$ 326.44 0.1)
After:	(1.38 $\boxed{-67.5}$ 326.44 0.1)

- Causes movement in the search space (local or global)
- Restores lost information to the population

Crossover: Recombination

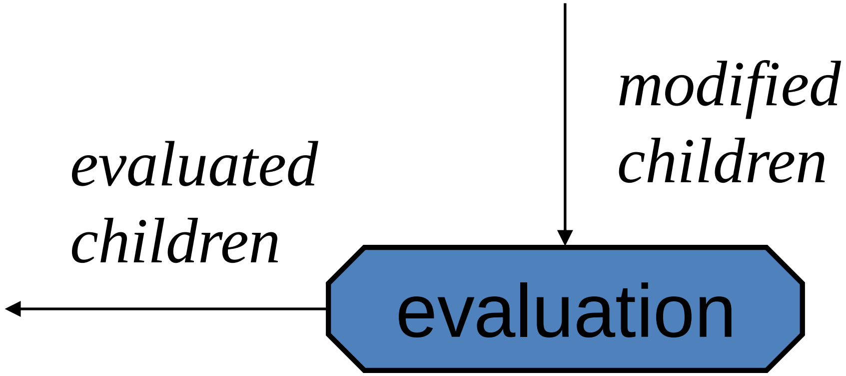
P1 (0 1 1 0 1 0 0 0) → (0 1 0 0 1 0 0 0) C1

P2 (1 1 0 1 1 0 1 0)   (1 1 1 1 1 0 1 0) C2

Crossover is a critical feature of genetic algorithms:

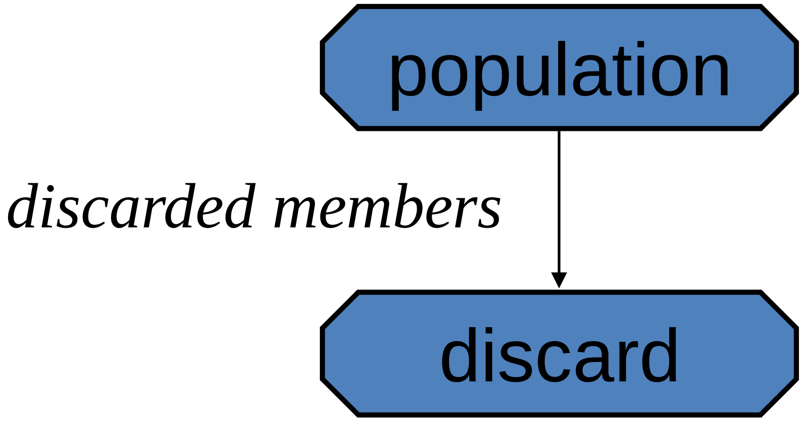
- It greatly accelerates search early in evolution of a population
- It leads to effective combination of schemata (subsolutions on different chromosomes)

Evaluation



- The evaluator decodes a chromosome and assigns it a fitness measure
- The evaluator is the only link between a classical GA and the problem it is solving

Deletion



- *Generational* GA:
  entire populations replaced with each iteration
- *Steady-state* GA:
  a few members replaced each generation

The Traveling Salesman Problem:

Find a tour of a given set of cities so that

- each city is visited only once
- the total distance traveled is minimized

Representation is an ordered list of city numbers known as an *order-based* GA.

1) London	3) Dunedin	5) Beijing	7) Tokyo
2) Venice	4) Singapore	6) Phoenix	8) Victoria

CityList1 (3 5 7 2 1 6 4 8)

CityList2 (2 5 7 6 8 1 3 4)

Crossover combines inversion and recombination:

Parent1	(3 5 7 $\boxed{2\ 1\ 6\ 4\ 8}$)
Parent2	(2 5 7 6 8 1 3 4)
Child	(2 5 7 $\boxed{2\ 1\ 6\ 3\ 4}$)

This operator is called the *Order1* crossover.

Mutation

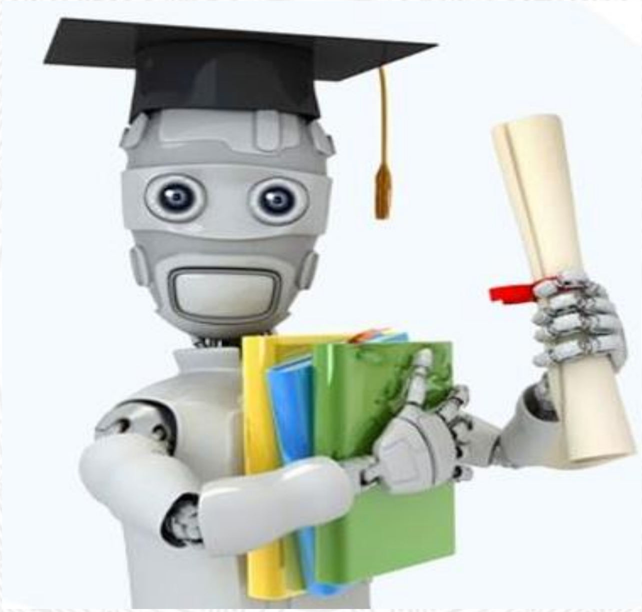
Mutation involves reordering of the list:

Before: (5 8 7 2 1 6 3 4)

After: (5 8 6 2 1 7 3 4)

SUMMARY

- Recap of EC metaphor
- Evolutionary Algorithm
- Genetic Algorithm



Enjoy the Course…!