8221010015
Evrimsel Hesaplama ve Uygulamaları

3.HAFTA

Evrimsel Algoritma Nedir?
EA Bileşenleri

2.Hafta - Tekrar

- Evrimsel Hesaplama Metaforu
- Evrimsel Hesaplama Tarihçesi ve Gelişimi
- Darwinist Evrim
- Adaptive Landscape Metaphor
- Doğal Genetik
  - Gen, Genom, Genetik Kod, Çaprazlama, Mutasyon
- Evrimsel Hesaplamanın Motivasyonları

EC Metafor Özet

- Sınırlı kaynakları olan bir ortamda bireylerden oluşan bir popülasyon vardır.
- Bu kaynaklar için ***rekabet***, çevreye daha iyi adapte olan ***seçkin*** bireylerin seçimine neden olmaktadır.
- Bu bireyler rekombinasyon ve mutasyon yoluyla yeni bireylerin üretimi için tohum görevi görür.
- Yeni bireyler uygunluklarını değerlendirilir ve hayatta kalmak için (muhtemelen ebeveynlerle de) yarışır.
- Zamanla ***Doğal Seleksiyon*** popülasyonun uygunluğunda bir artışa neden olur.

Evrimsel Algoritma Nedir?

- Evrimsel Algoritmalar «üret ve test et» algoritmaları kategorisine girer.

- Bunlar stokastik ve popülasyon tabanlı algoritmalardır.
  - İyi bireyler deterministik olarak belirlenmez.
  - Zayıf bireylerinde hayatta kalma şansı vardır.

- Varyasyon operatörleri (rekombinasyon ve mutasyon) gerekli çeşitliliği yaratır ve böylece yenilikçiliğe olanak sağlar.
  - Rekombinasyon bir ya da daha fazla bireye uygulanır, bir ya da daha fazla yeni birey elde edilir.
  - Mutasyon bir bireye uygulanır ve yeni bir birey elde edilir.

- Seçim, popülasyondaki çözümlerin ortalama kalitesini arttırmak için bir güç olarak rol oynar.

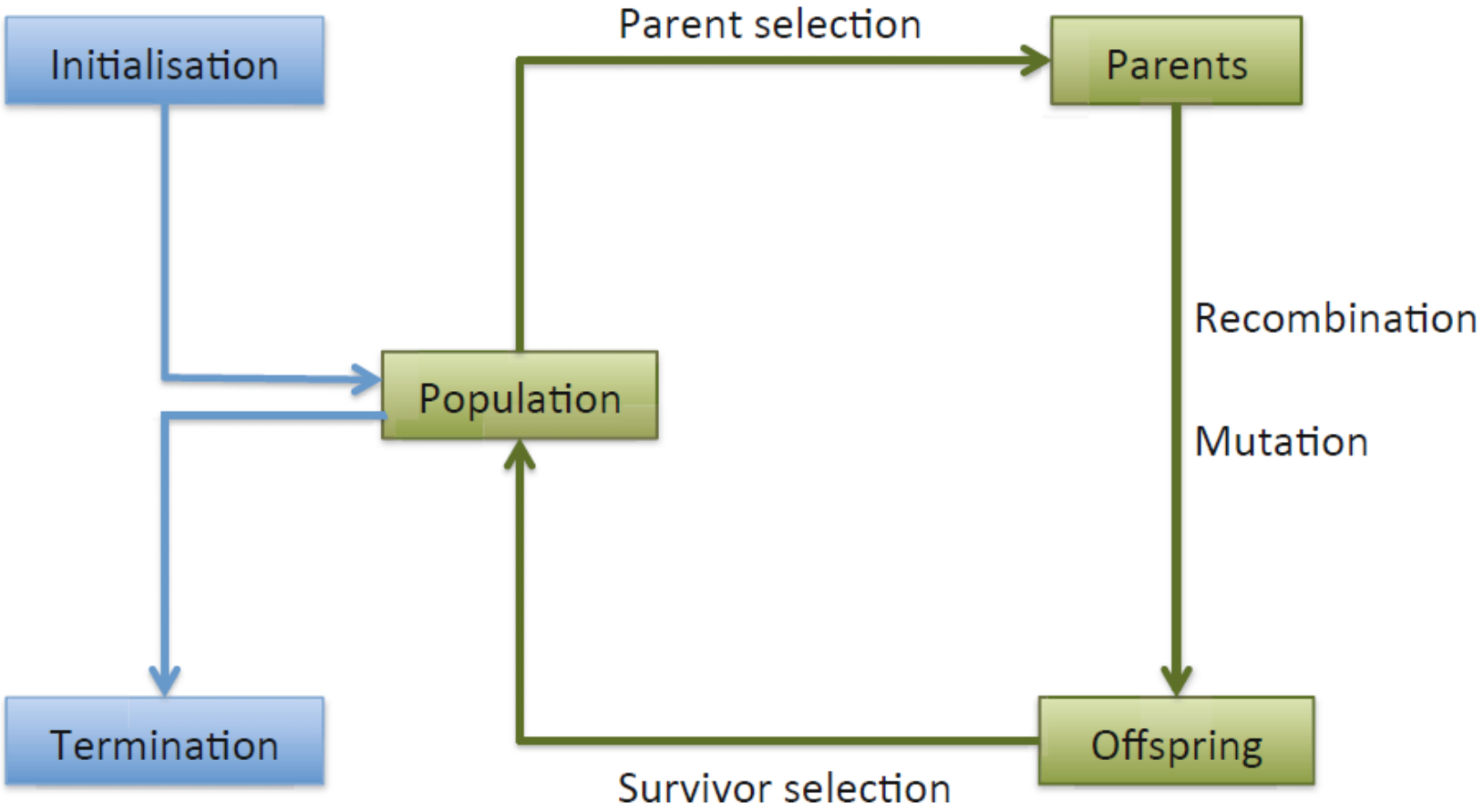
Evrimsel Algoritmanın Genel Şeması

```
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
```

Evolutionary Algorithm

- It is easy to see that this scheme falls into the category of generate-and-test algorithms.
- The evaluation (fitness) function provides a heuristic estimate of solution quality
- The search process is driven by the variation and selection operators.
- Evolutionary algorithms possess a number of features that can help position them within the family of generate-and-test methods:
  - EAs are population based, i.e., they process a whole collection of candidate solutions simultaneously.
  - Most EAs use recombination, mixing information from two or more candidate solutions to create a new one.
  - EAs are stochastic.

Evrimsel Algoritma Akış Şeması



The two pillars of evolution

There are two competing forces active

- Increasing population diversity by genetic operators
  - mutation
  - recombination

Push towards novelty

- Decreasing population diversity by selection
  - of parents
  - of survivors

Push towards quality

Evrimsel Algoritmaların Bileşenleri

- En önemli bileşenler:
  - representation (definition of individuals)
  - evaluation function (or fitness function)
  - population
  - parent selection mechanism
  - variation operators, recombination and mutation
  - survivor selection mechanism (replacement)

- Tam ve çalışabilir algoritma için her bir bileşenin belirlenmesi ve başlangıç prosedürünün tanımlanması gerekir.

- Ayrıca algoritmanın belirli bir aşamada durmasını istiyorsak durdurma kriteri (termination condition) sağlanmalıdır.

Representation (Definition of Individuals)

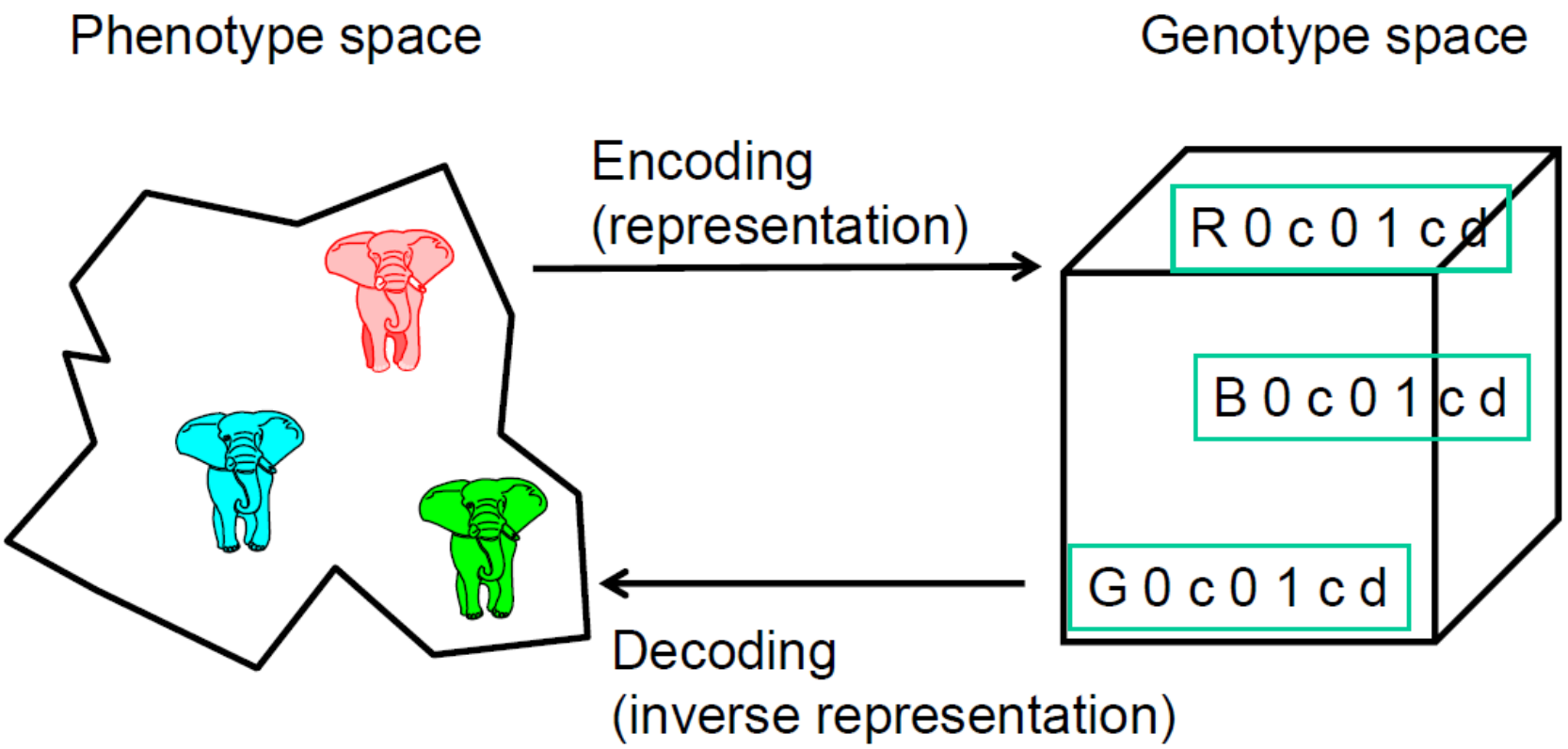
- Bridge – to link the real world to EA world
- Undertaken by domain experts
  - To decide how possible solutions should be specified and stored in a way that can be manipulated by a computer.
- Leads to two levels of existence
  - phenotype: object in original problem context, the outside
  - genotype: code to denote that object, the inside (chromosome, "digital DNA")

- This first design step → Representation
  - A mapping from the phenotypes onto a set of genotypes that are said to represent them
- Implies two mappings: (18 ←→ 10010)
  - Encoding : phenotype=> genotype (not necessarily one to one)
  - Decoding : genotype=> phenotype (must be one to one)

Representation

- The evolutionary computation literature contains many synonyms:
- On the side of the original problem context the terms **candidate solution**, phenotype, and **individual** are all used to denote possible solutions.
  - The space of all possible candidate solutions is commonly called the **phenotype space**.
- On the side of the EA, the terms genotype, **chromosome**, and again individual are used to denote points in the space where the evolutionary search actually takes place.
  - This space is often termed the **genotype space**.
- There are also many synonymous terms for the elements of individuals. A placeholder is commonly called a variable, a **locus** (plural: loci), a position, or – in a biology-oriented terminology – a **gene**.
  - An object in such a place can be called a value or an **allele**.

Representation



Fitness Function (Evaluation)

- Represents the task to solve, the requirements the population should adapt to meet (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
- *objective* function or *quality* 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

Population

- 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

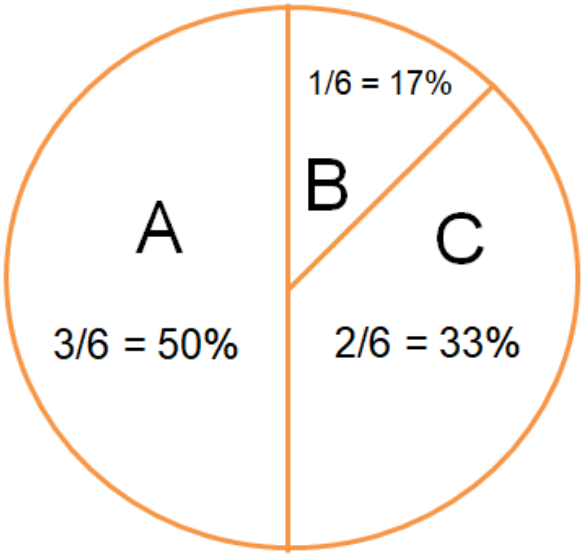
- Diversity of a population refers to the number of different fitnesses / phenotypes / genotypes present (note: not the same thing)

Parent Selection Mechanism

- 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

Parent Selection Mechanism

- 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

Variation Operators (Mutation and Recombination)

- To create new individuals from old ones
- Generation step
  - Generate-and-Test search perspective
- Variation operators in EC are divided into two types based on their arity, distinguishing
  - Unary (mutation)
  - n-ary versions (recombination)

Variation Operators- Mutation

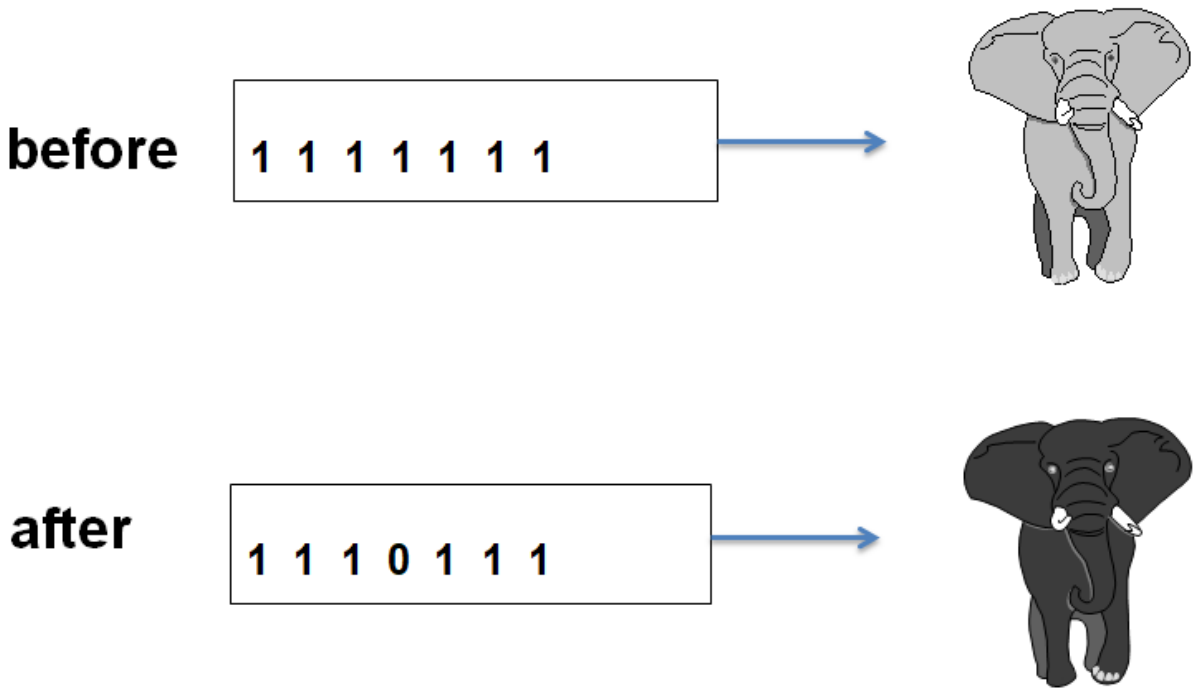
- Applied to one genotype and delivers a (slightly) modified mutant, the **child** or **offspring**.

- Always stochastic
  - its output (the child) depends on the outcomes of a series of random choices.

- Not all unary operators are seen as mutation

- Importance ascribed depends on representation and historical dialect:
  - GAs – background operator responsible for preserving and introducing diversity (fresh blood)
  - EP for FSM's / continuous variables – only search operator
  - GP – hardly used

Variation Operators- Mutation



Variation Operators- Mutation

- Yeni bir çocuk üretmek arama uzayında yeni bir noktaya adım atmak demektir.
- Mutasyonun alan bağımlı olduğu garanti edilebilir.
- EA ‘ya yeterli süre sağlandığında verilen problemin global optimumunu keşfedeceğini iddia eden bazı teoremler vardır.
- Bu bir olası çözümü temsil eden her genotipe, varyasyon operatörleri ile ulaşabileceği bağlantısına dayanır.
- Bu koşulu sağlamanın en basit yolu, mutasyon operatörünün her yere sıçramasına izin vermektir.
  - Örneğin, herhangi bir alelinin sıfır olmayan bir olasılık ile herhangi bir diğerine mutasyona uğramasına izin vererek.
- Bununla birlikte, birçok araştırmacı bu kanıtların sınırlı pratik öneme sahip olduğunu ve EA uygulamalarının genellikle bu özelliğe sahip olmadığını düşünmektedir.

Variation Operators- Recombination

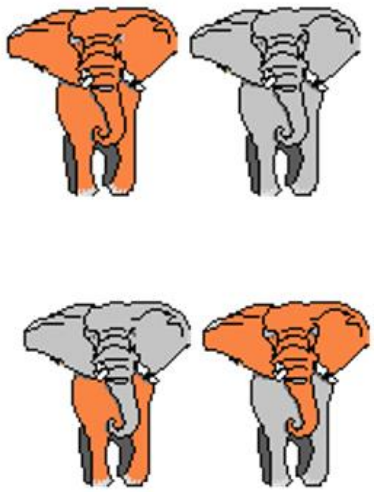
- A binary variation operator is called **recombination** or **crossover**.
  - Merges information from two parent genotypes into one or two offspring genotypes.
- Like mutation, recombination is a stochastic operator
  - the choices of what parts of each parent are combined, and how this is done, depend on random drawings.
- The role of recombination differs between EC dialects:
  - GP - it is often the only variation operator,
  - GA - it is seen as the main search operator,
  - EP - it is never used.
- Recombination operators with a higher arity (using more than two parents) are mathematically possible and easy to implement, but have no biological equivalent.
- Perhaps this is why they are not commonly used, although several studies indicate that they have positive effects on the evolution.

Variation Operators- Recombination

- 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
- Variation operators are representation-dependent.

Chromosome1	11011|00100110110
Chromosome2	11011|11000011110
Offspring1	11011|11000011110
Offspring2	11011|00100110110

Single Point Crossover



Survivor Selection Mechanism (Replacement)

- Survivor selection or ***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)

Initialisation and Termination

- Initialisation
  - First population is seeded by randomly generated individuals.
  - Problem-specific heuristics can be used in this step, to create an initial population with higher fitness.
  - Whether this is worth the extra computational effort, or not, very much depends on the application at hand.
- Termination
  - Optimum is known or not known
  - EAs are stochastic and mostly there are no guarantees of reaching such an optimum
  - So this condition might never get satisfied, and the algorithm may never stop
  - Needs to a certain stop condition

Termination Condition

- Options for termination condition
  - The maximally allowed CPU time elapses.
  - The total number of fitness evaluations reaches a given limit.
  - The fitness improvement remains under a threshold value for a given period of time (i.e., for a number of generations or fitness evaluations).
  - The population diversity drops under a given threshold.
- Technically, the actual termination criterion in such cases is a disjunction: optimum value hit or condition X satisfied.

What are the different types of EAs?

- Historically different flavours of EAs have been associated with different data types to represent solutions
  - Binary strings : Genetic Algorithms
  - Real-valued vectors : Evolution Strategies
  - Finite state Machines: Evolutionary Programming
  - LISP trees: Genetic Programming

- These differences are largely irrelevant, best strategy
  - choose representation to suit problem
  - choose variation operators to suit representation

- Selection operators only use fitness and so are independent of representation

An Evolutionary Cycle by Hand

- Problem
  - Maximising the values of x^2 for integers in the range 0–31.
- Representation
  - Five-bit binary encoding
    - Integers (phenotype) → bit-strings (genotype)
- Parent Selection
  - Fitness proportional mechanism

$$p_i = f(i) / \sum_{j \in P} f(j)$$

An Evolutionary Cycle by Hand

- Survivor Selection Mechanism
  - All existing individuals are removed from the population
  - All new individuals are added to it without comparing fitness values.

- Variation Operator –Mutation
  - Executed by generating a random number (from a uniform distribution over the range [0, 1]) in each bit position,
  - Comparing it to a fixed threshold, usually called the mutation rate.
  - If the random number is below that rate, the value of the gene in the corresponding position is flipped.

- Variation Operator – Recombination
  - Classic one-point crossover
  - This operator is applied to two parents and produces two children by choosing a random crossover-point along the strings
  - Swapping the bits of the parents after this point

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

Table 3.1. The x^2 example, 1: initialisation, evaluation, and parent selection

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

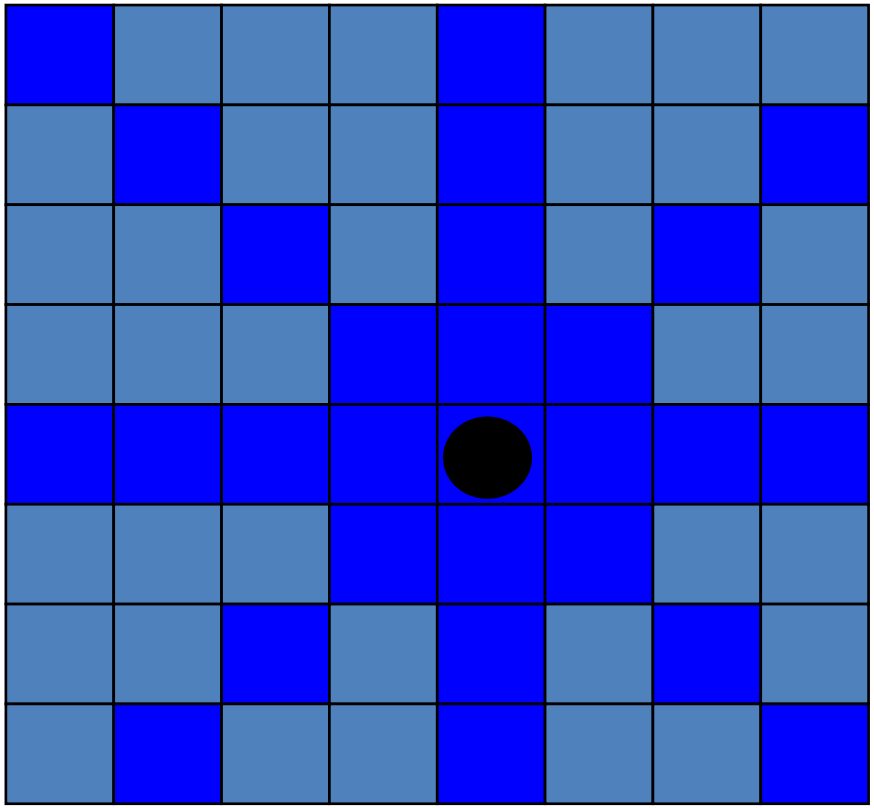
Table 3.2. The x^2 example, 2: crossover and offspring evaluation

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

Table 3.3. The x^2 example, 3: mutation and offspring evaluation

- The mutations shown happen to have caused positive changes in fitness
- But we should emphasise that in later generations, as the number of 1's in the population rises, mutation will be on average (but not always) deleterious.
- Although manually engineered, this example shows a typical progress:
  - The average fitness grows from 293 to 588.5,
  - The best fitness in the population from 576 to 729 after crossover and mutation.

Example:
The 8-queens problem

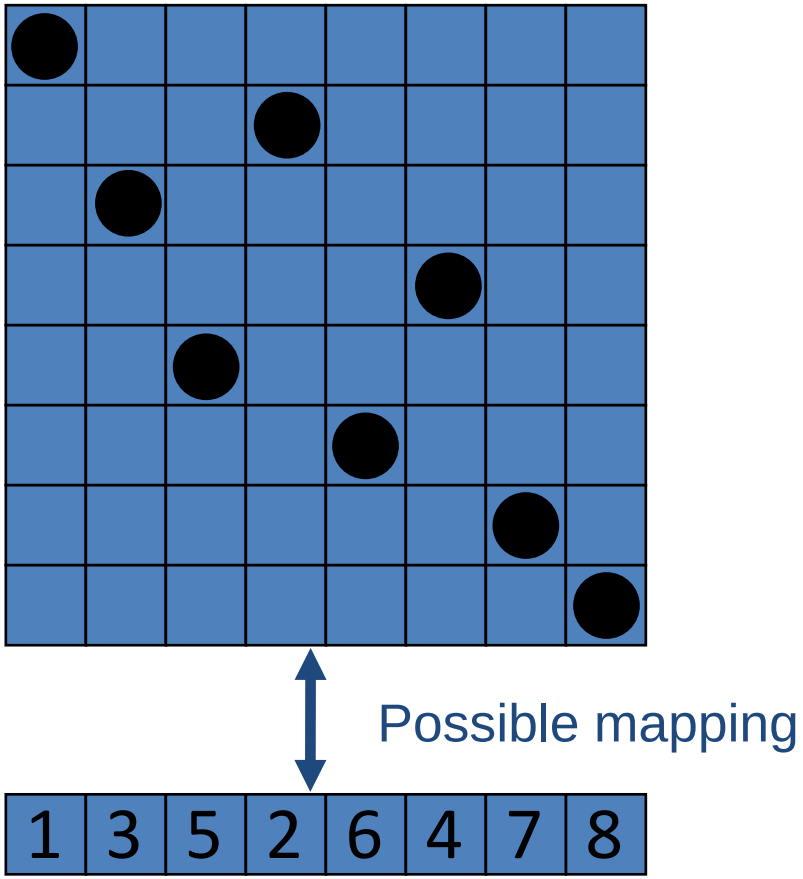


Place 8 queens on an 8x8 chessboard in such a way that they cannot check each other

The 8-queens problem: Representation

Phenotype:
a board configuration

Genotype:
a permutation of
the numbers 1–8



The 8-queens problem: Fitness evaluation

- Penalty of one queen: the number of queens she can check
- Penalty of a configuration: the sum of penalties of all queens
- Note: penalty is to be minimized
- Fitness of a configuration: inverse penalty to be maximized

The 8-queens problem: Mutation

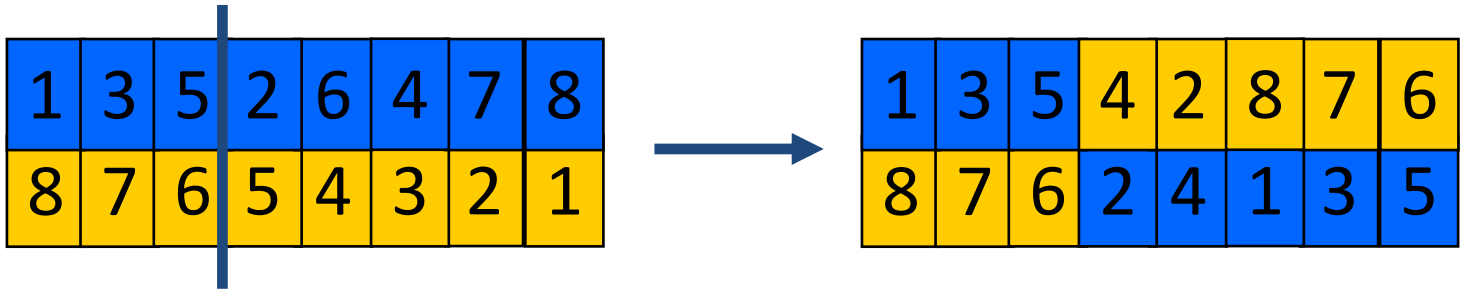
Small variation in one permutation, e.g.:

- swapping values of two randomly chosen positions,

The 8-queens problem: Recombination

Combining two permutations into two new permutations:

- choose random crossover point
- copy first parts into children
- create second part by inserting values from other parent:
  - in the order they appear there
  - beginning after crossover point
  - skipping values already in child



The 8-queens problem: Selection

- Parent selection:
  - Pick 5 parents and take best two to undergo crossover
- Survivor selection (replacement)
  - When inserting a new child into the population, choose an existing member to replace by:
  - sorting the whole population by decreasing fitness
  - enumerating this list from high to low
  - replacing the first with a fitness lower than the given child

The 8-queens problem: Summary

Representation	Permutations
Recombination	"Cut-and-crossfill" crossover
Recombination probability	100%
Mutation	Swap
Mutation probability	80%
Parent selection	Best 2 out of random 5
Survival selection	Replace worst
Population size	100
Number of Offspring	2
Initialisation	Random
Termination condition	Solution or 10,000 fitness evaluation

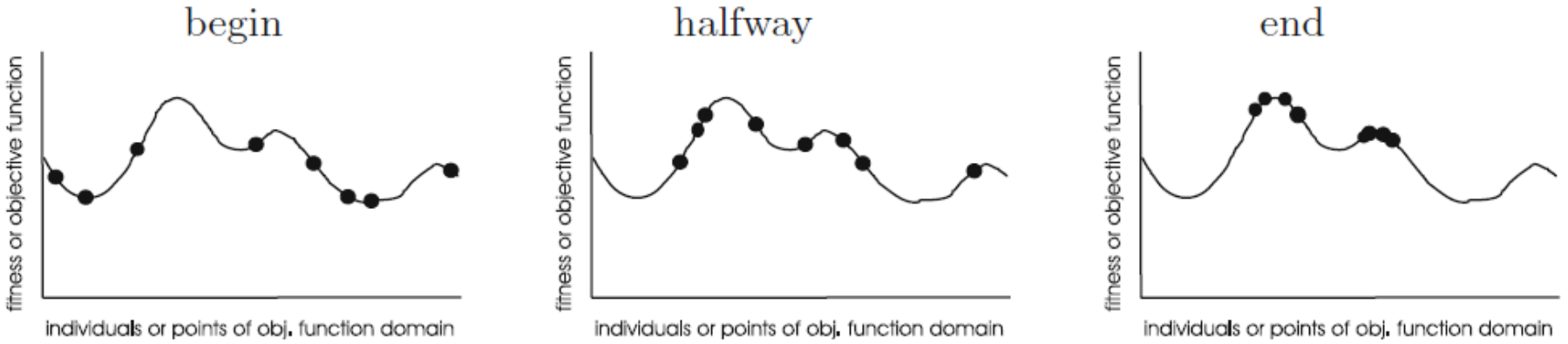
Note that is ***only one possible***
set of choices of operators and parameters

Knapsack Problem
Description of the EA

- We are given a set of *n* items, each of which has attached to it some value *vi*, and some cost *ci*.

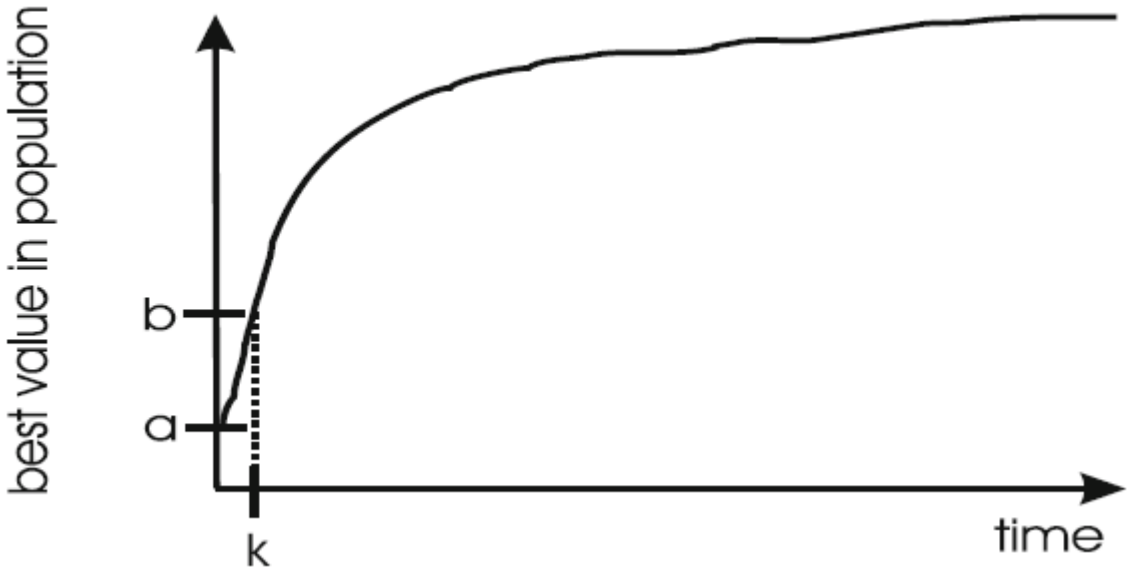
Representation	Binary strings of length n
Recombination	One-point crossover
Recombination probability	70%
Mutation	Each value inverted with independent probability p_m
Mutation probability p_m	$1/n$
Parent selection	Best out of random 2
Survival selection	Generational
Population size	500
Number of offspring	500
Initialisation	Random
Termination condition	No improvement in last 25 generations

Typical Progress of an EA



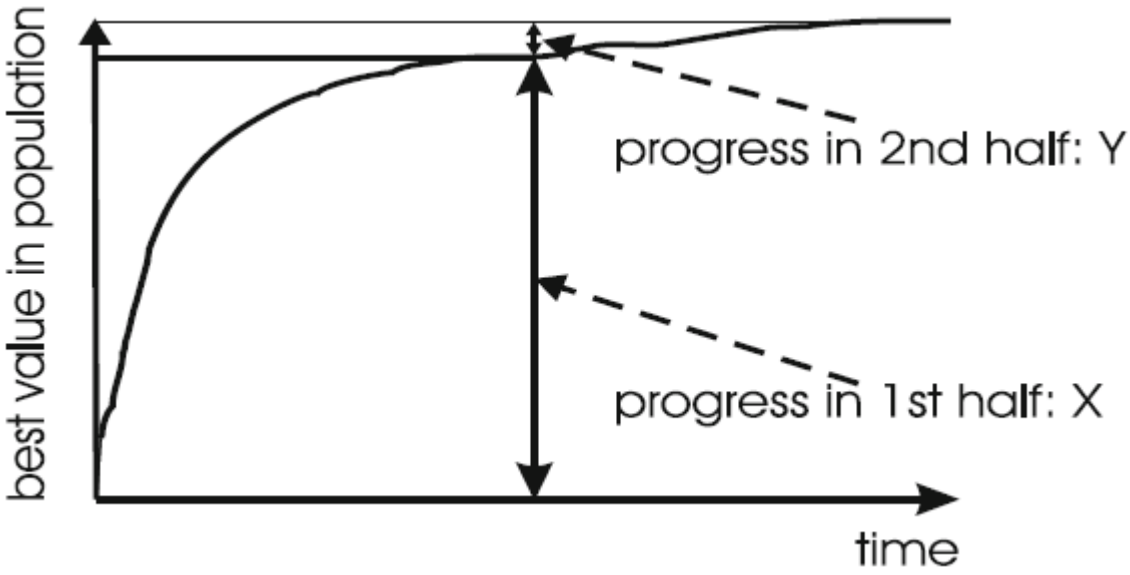
- Trade-off between exploration and exploitation
  - **Exploration** is the generation of new individuals in as-yet untested regions of the search space.
  - **Exploitation** means the concentration of the search in the vicinity of known good solutions.
- Too much of the former can lead to inefficient search, and too much of the latter can lead to a propensity to focus the search too quickly
- **Premature convergence** is the well-known effect of losing population diversity too quickly, and getting trapped in a local optimum.

The Operation of an EA



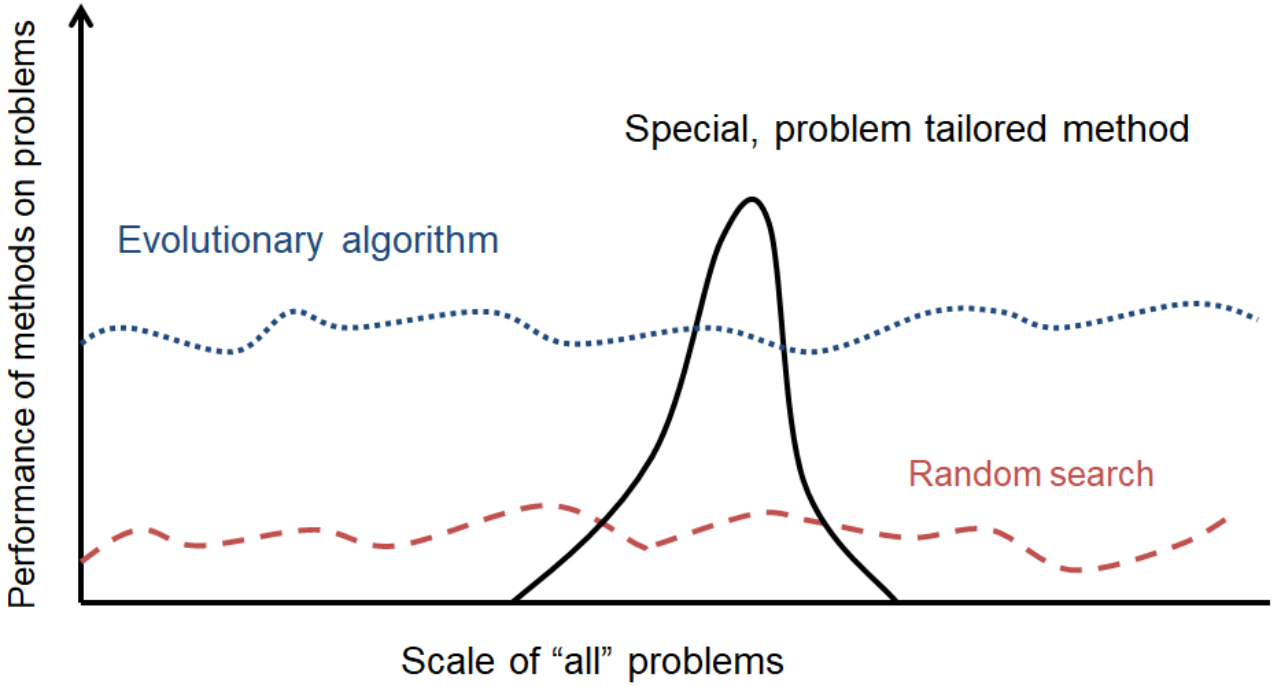
- The other effect we want to illustrate is the **anytime behaviour** of EAs by plotting the development of the population's best fitness value over time
- Illustration of why heuristic initialisation might not be worth additional effort.
- Level *a* shows the best fitness in a randomly initialised population; level *b* belongs to heuristic initialisation

The Operation of an EA



- Why long runs might not be worth performing
- *X* shows the fitness increase in the first half of the run, while *Y* belongs to the second half

Typical EA behaviour: EAs as problem solvers: Goldberg view (1989)



- No Free Lunch theorem
- No Free Lunch theorem has shown that (under some conditions) no black-box algorithm can outperform random walk when averaged over 'all' problems [467].
- That is, showing the EA line always above that of random search is fundamentally incorrect.

	Natural evolution	Artificial evolution
Fitness	Observed quantity: *a posteriori* effect of selection ('in the eye of the observer').	Predefined *a priori* quantity that drives selection.
Selection	Complex multifactor force based on environmental conditions, other individuals of the same species and other species (e.g., predators). Viability is tested continually; reproducibility is tested at discrete times.	Randomized operator with selection probabilities based on given fitness values. Parent selection and survivor selection both happen at discrete times.
Genotype-phenotype mapping	Highly complex biochemical process influenced by the environment.	Relatively simple mathematical transformation or parameterised procedure.
Variation	Offspring created from one (asexual reproduction) or two parents (sexual reproduction).	Offspring may be generated from one, two, or many parents.
Execution	Parallel, decentralized execution; birth and death events are not synchronised.	Typically centralized with synchronised birth and death.
Population	Spatial embedding implies structured populations. Population size varies according to the relative number of death and birth events.	Typically unstructured and panmictic (all individuals are potential partners). Population size is kept constant by synchronising time and number of birth and death events.

Table 3.6. Differences between natural and artificial evolution

Kaynaklar

- A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Natural Computing Series, Springer