IV. Evolutionary algorithms

The evolutionary algorithm is a search system

- It handles a population of the elements (individuals) of the problem space at the same time.
- Firstly an initial population is selected mostly at random. The aim is to find either the best individual or a better population.
- In every turn the better individuals are selected for reproduction and the population is updated with their offspring. This change of the population is an irreversible step, so the evolution is controlled by an irrevocable strategy.
- An individual is better than the other if it is somehow closer to the correct answer. The fitness function can measure this „distance”.

Evolutionary operators

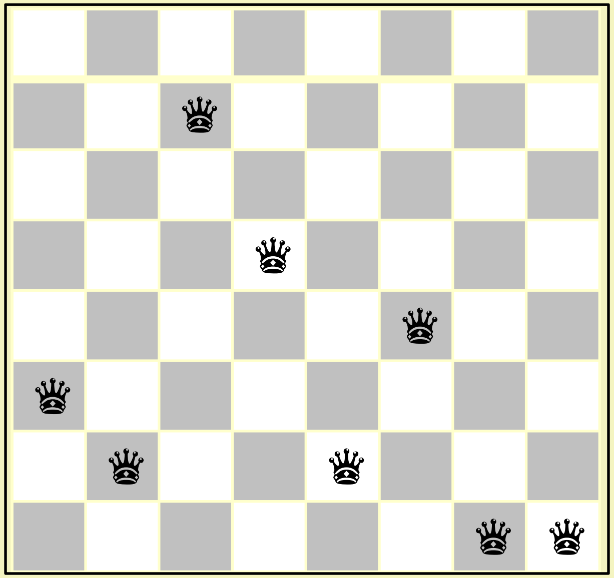
- *Selection*: individuals are selected for reproduction
- *Recombination (crossover)*: pairs of the selected individuals (parents) are mated to create their offspring
- *Mutation*: offspring can be changed a bit with a small independent property
- *Replacement*: a new population is constructed from the elder one and the mutated offspring

DATA *:= initial value*
while ¬ *termination condition*(DATA) **loop**
    SELECT R FROM *rules that can be applied*
    DATA *:=* R(DATA)
endloop

Evolutionary algorithm

Procedure EA
population := initial population
while *not termination condition* ***loop***
    parents   := selection(*population*)
    offspring:= recombination(*parents*)
    offspring := mutation(*offspring*)
    population := replacement(*population* , *offspring*)
endloop

n-queens problem 1



3	2	7	5	2	4	1	1

fitness value: 23

- Individual: a possible placement of the queens where each column contains exactly one queen
- Representation: the sequence of the row positions of the 8 queens
- Fitness function: number of nonattacking pairs of queens

Evolutionary cycle

selection

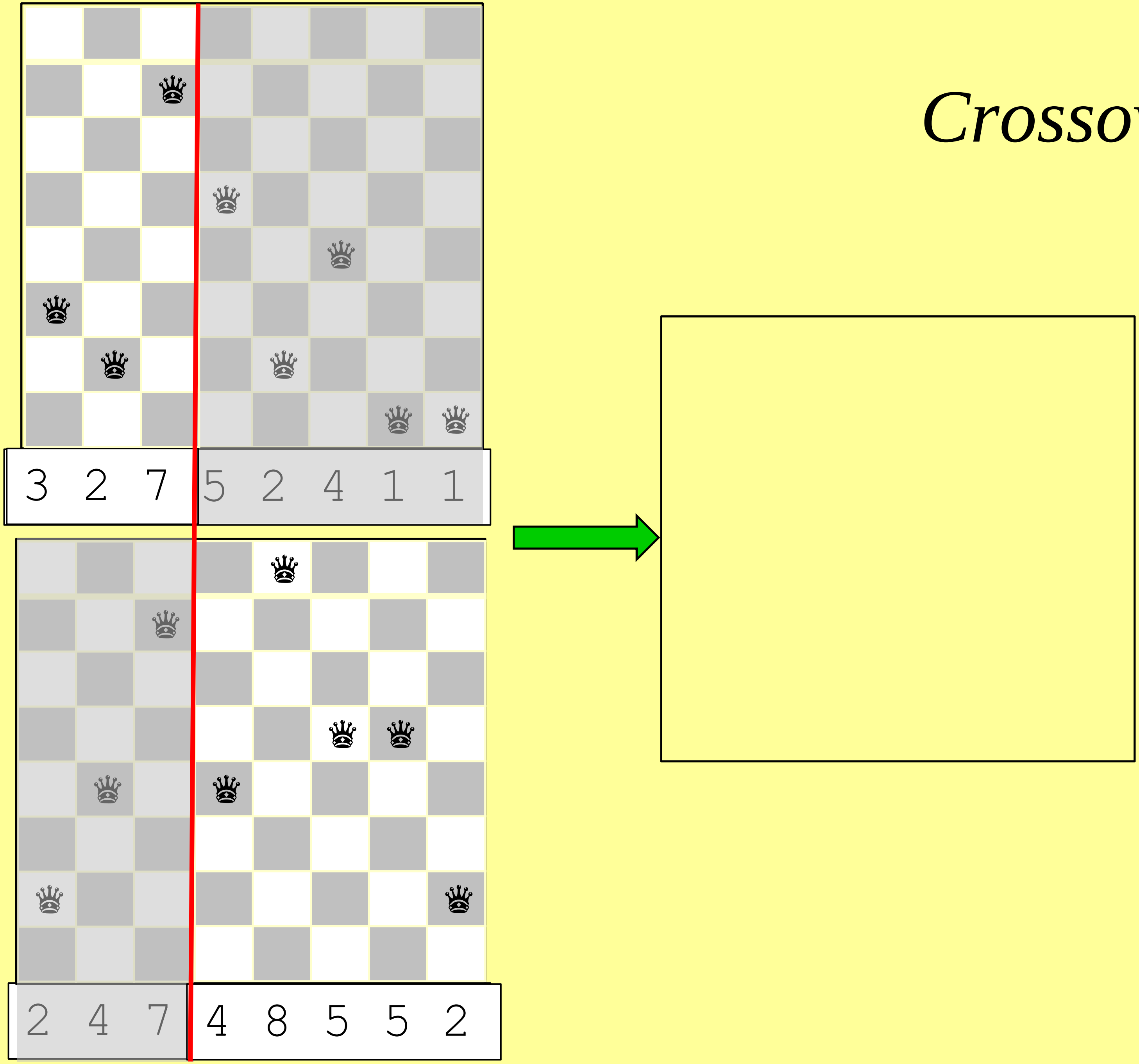
24748552 ~ 24
32752411 ~ 23
24415124 ~ 20
32543213 ~ 11

247|48552
327|52411
32752|411
24415|124

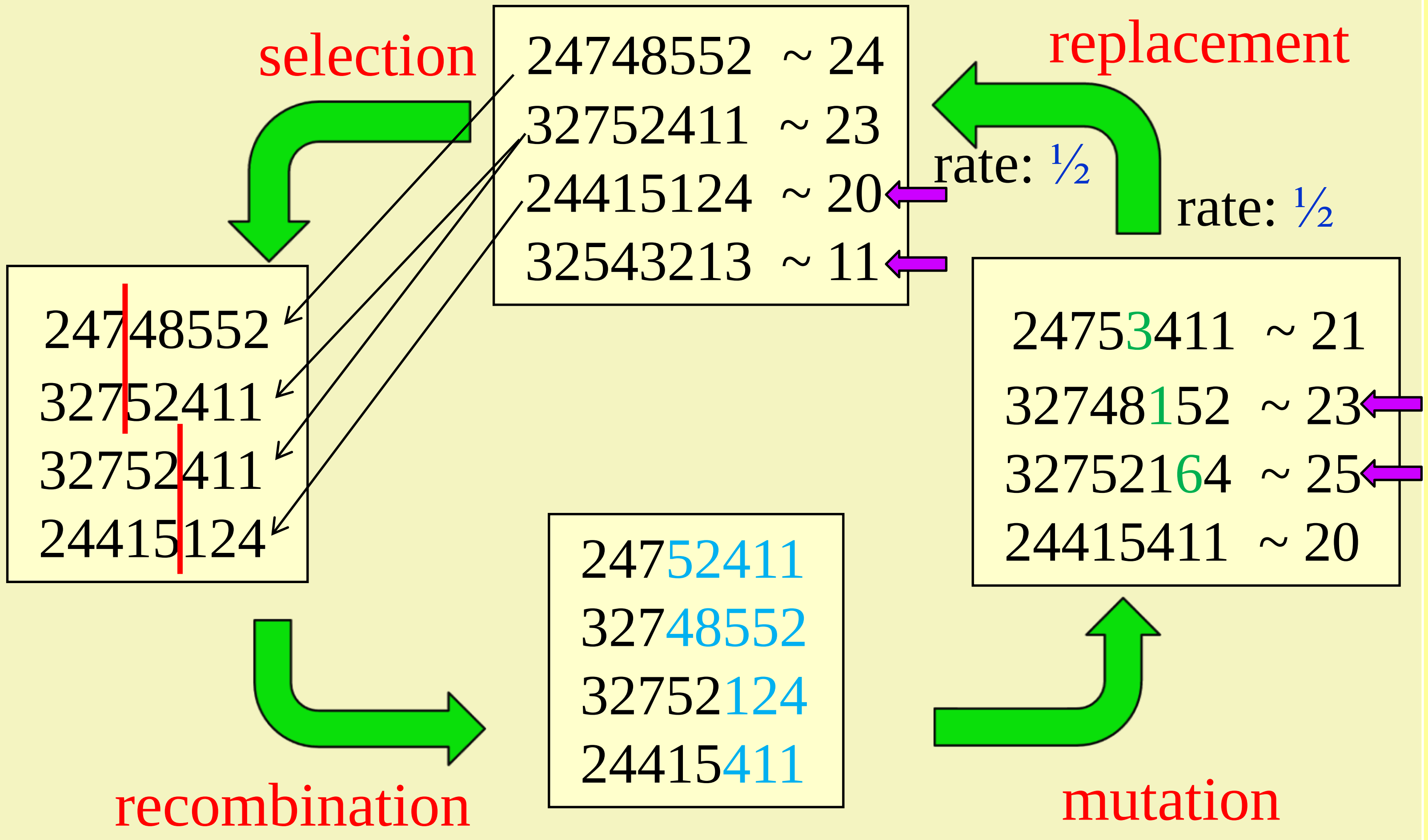
recombination

24752411
32748552
32752124
24415411

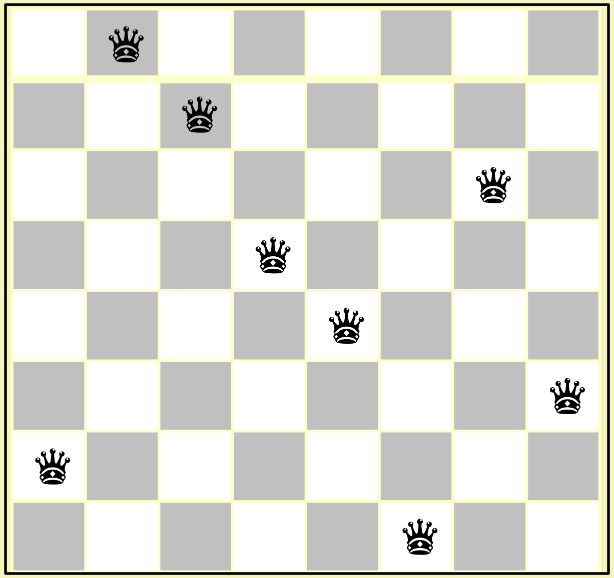
Crossover



Evolutionary cycle



n-queens problem 2

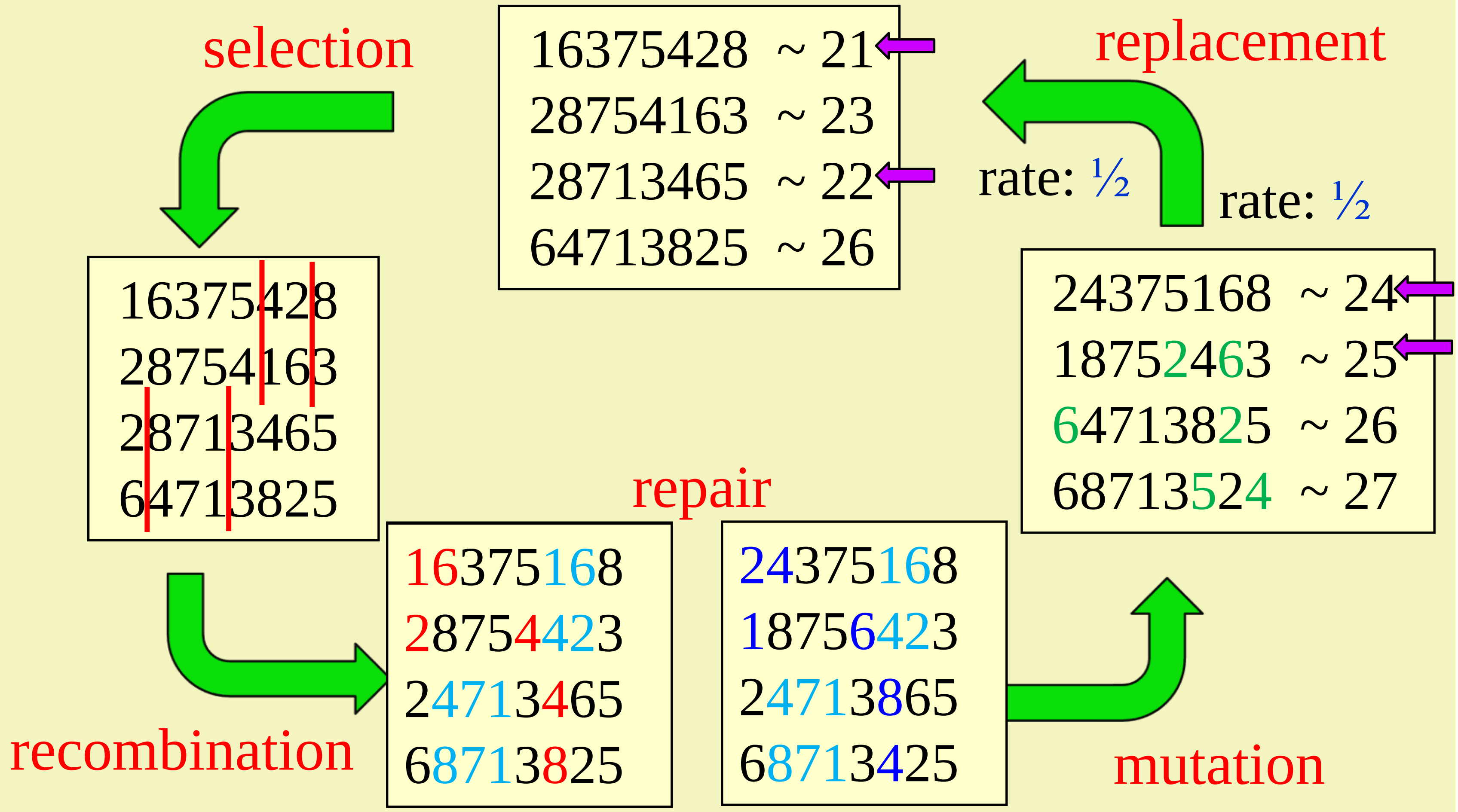


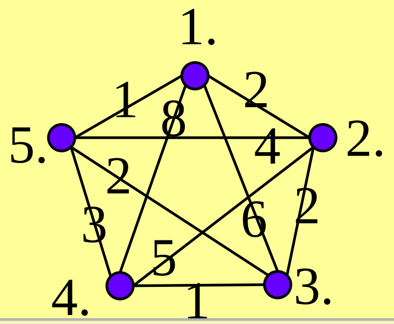
2	8	7	5	4	1	6	3

fitness value: 23

- Individual: a possible placement of the queens where each column and each row contains exactly one queen
- Representation: permutation of the row positions
- Fitness function: number of nonattacking pairs of queens

Evolutionary cycle



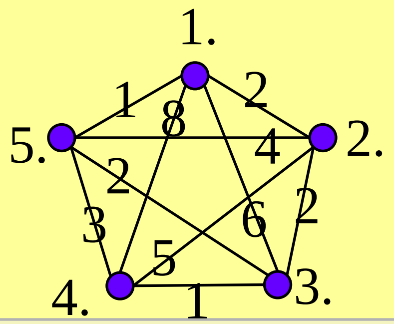


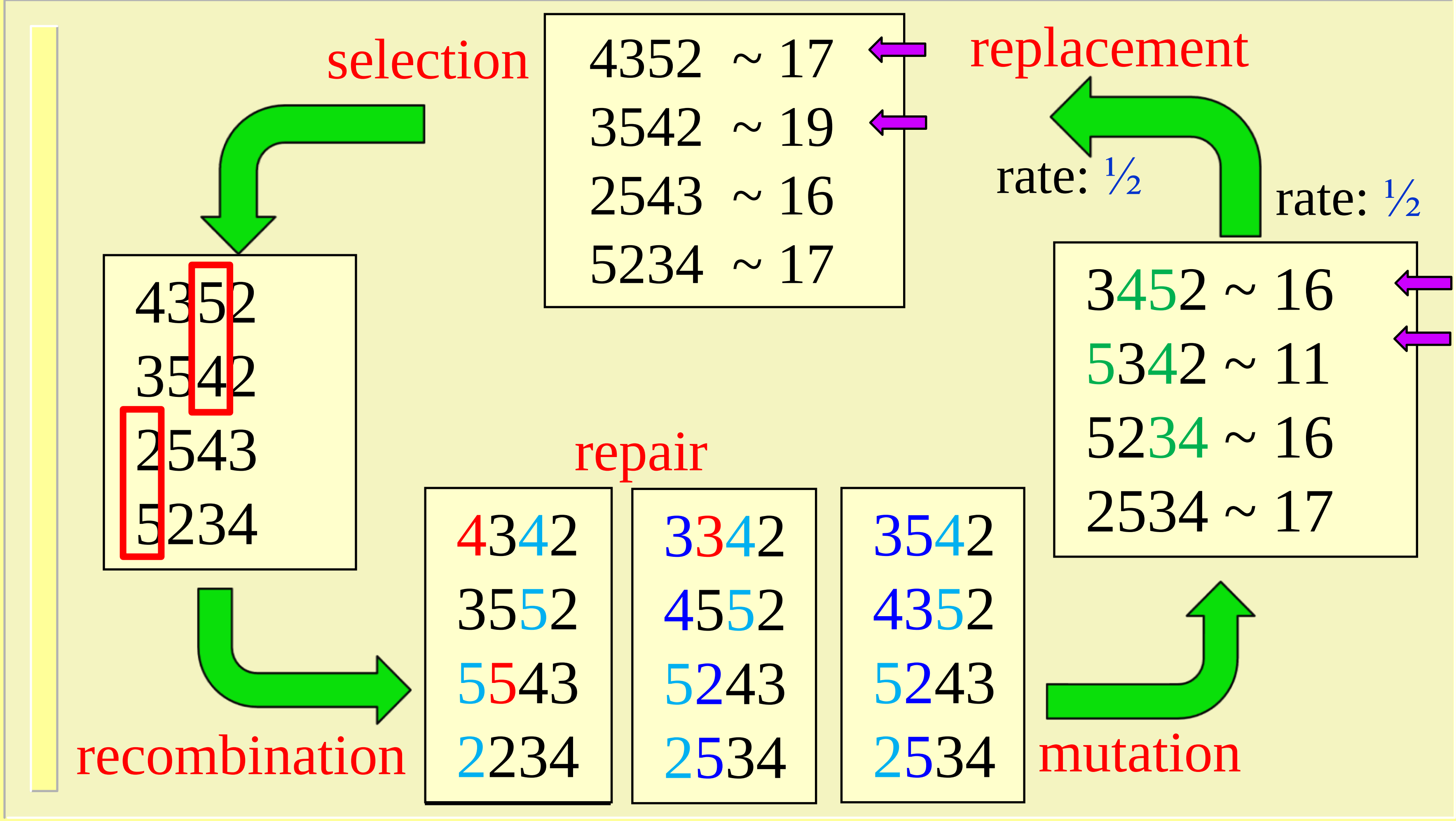
Traveling salesman problem

The traveling salesman must visit every city in his territory exactly once and then return to home city covering the shortest distance. (All distances between cities are known)

- Individual : one tour
- Representation: permutation of the cities (numbers) without the home city
- Fitness function: cost of the tour

Evolutionary cycle





Satisfiability problem (SAT)

There is a given Boolean statement in CNF with n Boolean variables. Find a vector of truth assignments for all n variables so that the formula be true.

E.g.: $(x_1 \vee \neg x_2 \vee x_5) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_4) \wedge (\neg x_2 \vee x_5)$

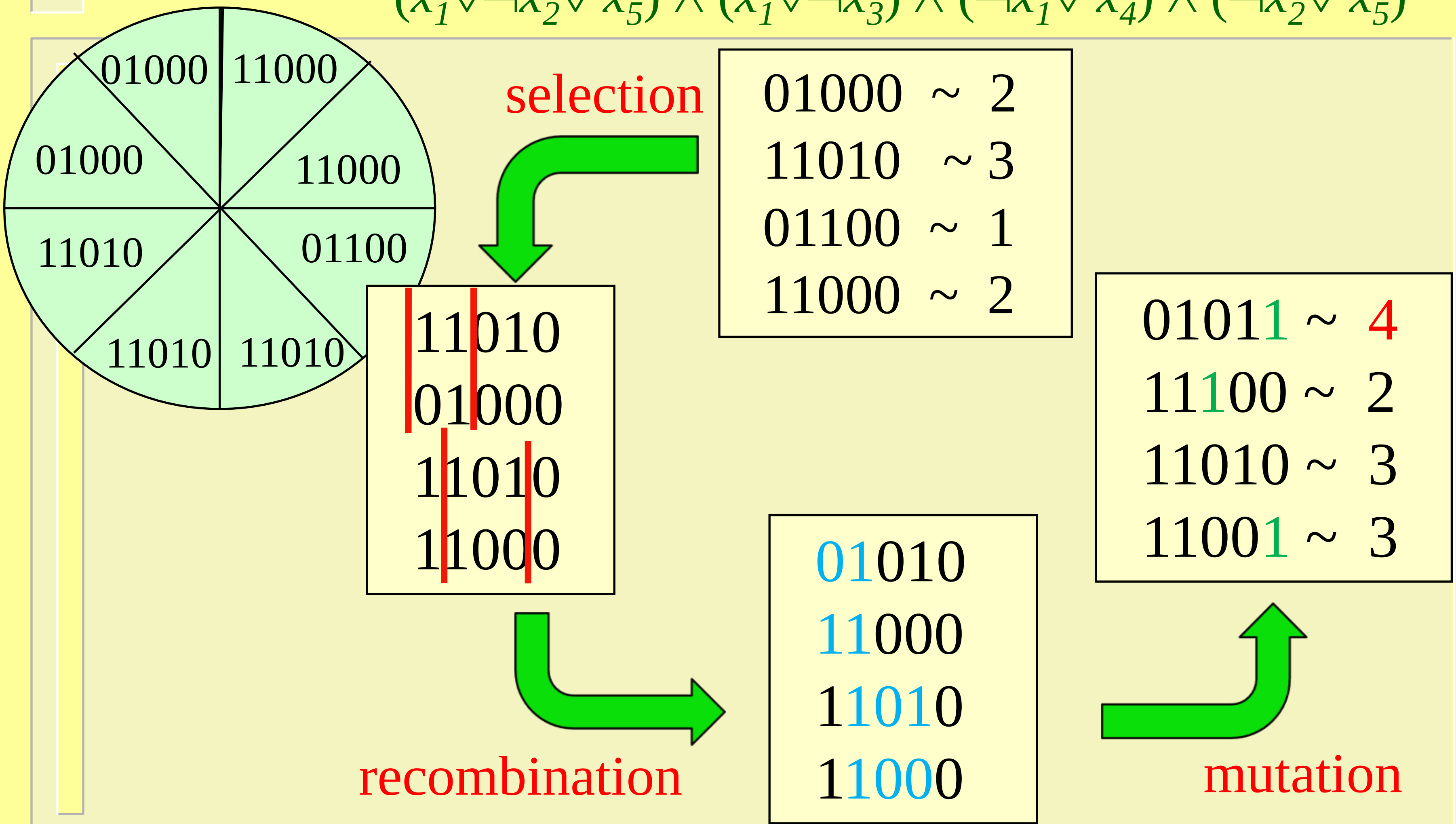
one solution: $x_1 = true, x_2 = any, x_3 = any, x_4 = true, x_5 = true$

- Individual : a possible truth assignment
- Representation: a sequence of bits
- Fitness function: the number of the clauses of the formula that has got true value in the given truth assignment

Evolutionary cycle

Roulette wheel

$(x_1 \lor \neg x_2 \lor x_5) \land (x_1 \lor \neg x_3) \land (\neg x_1 \lor x_4) \land (\neg x_2 \lor x_5)$



Elements of the evolutionary algorithm

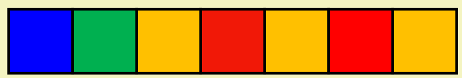
- Model: individuals and their representation
- Fitness function (heuristics)
- Evolutionary operators
  - selection, recombination, mutation, replacement
- Initial population, termination condition (goal)
- Settings of the strategy parameters
  - size of the population, probability of mutation, rate of the offspring, rate of the replacement

Representation

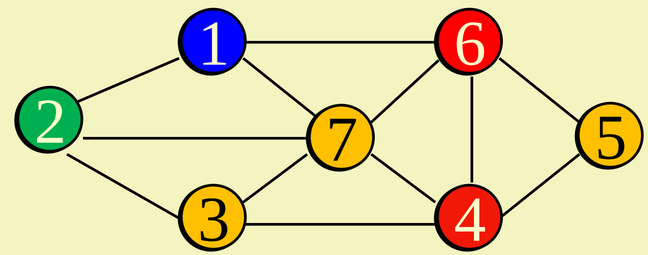
- Most commonly an individual is represented by a sequence of signals, that is, a string over a finite alphabet (chromosome).
- A signal or a section of signals with its position in the code (gene) can describe one property of the individual.
  - Thus the structure of the code can be dismembered property by property. If one signal is changed, then one property of the individual is changed a bit, too.
- Frequently coding methods:
  - Vector: fix length sequence of real numbers or integers
  - Binary code: fix length sequence of bits
  - Permutation of finite elements

Representations and fitness functions of graph coloring problem

There is a given simple finite undirected graph. How can its vertices be colored so that no two adjacent vertices share the same color?

Direct encoding 

1\. 2. 3. 4. 5. 6. 7.



$x[i]$ is the color of the i^{th} vertex.

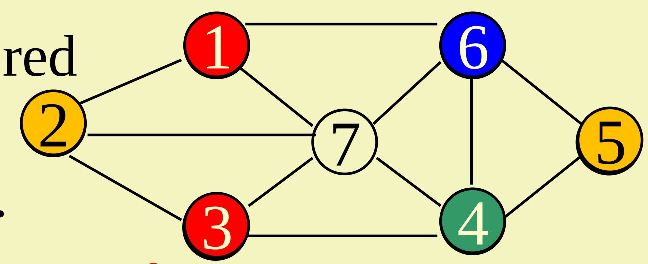
f is the number of the correct edges.

f = 10

Indirect encoding

2	5	1	3	4	6	7

In the i^{th} step the $x[i]^{th}$ vertex must be colored by the possible lightest color regarding the colors of neighbor vertices, if possible.



f is the number of the colored vertices.

f = 6

Representation and fitness functions of problem of rock-paper-scissors

Find a good strategy for the rock-paper-scissors world championship!

- This sort of strategy should give our next step regarding with a few (say two) previous combats.
  - For example:

    Me: R P *Suggestion*: R

    Op: S S
  - This is just one slice of the total strategy. We need all suggestions for all possible antecedents.

Representation

Code of one individual: $\{0,1,2\}^{0..80}$

Signal

R ~ 0
P ~ 1
S ~ 2

Strategy (individual): 1201 … 20

Antecedent (MeOpMeOp)	*Suggestion*
RRRR ~ 0000 ~ 0	P ~ 1
RRRP ~ 0001 ~ 1	S ~ 2
RRRS ~ 0002 ~ 2	R ~ 0
…	
PSRP ~ 1201 ~ 46	P ~ 1
…	
SSSP ~ 2221 ~ 79	S ~ 2
SSSS ~ 2222 ~ 80	R ~ 0

Fitness function

Individual: 1 2 0 1 ... 2 0

Signal

R ~ 0

P ~ 1

S ~ 2

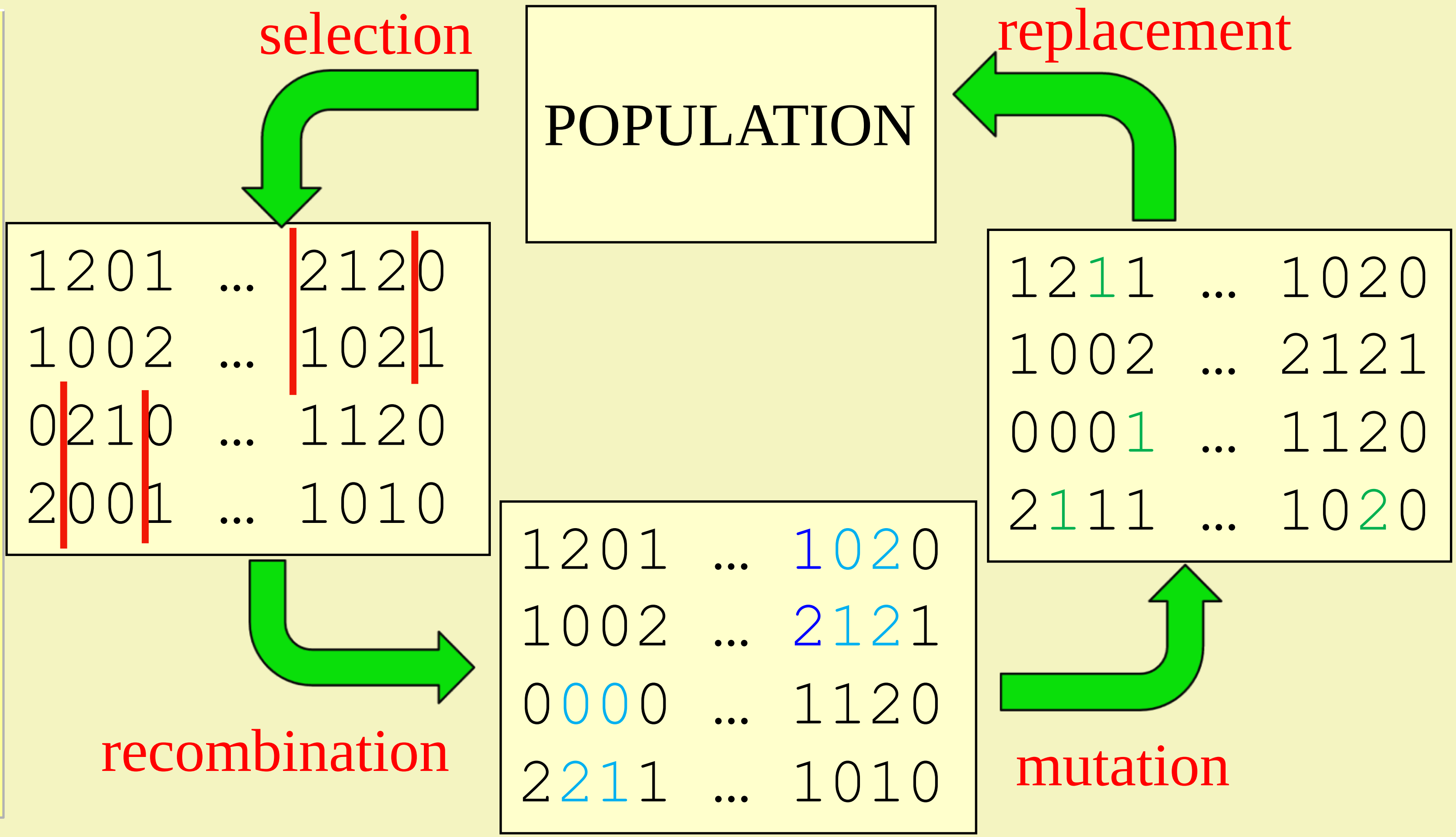
Specimen battle:

Me: 0 0 0 2 2 2 1 2 2 2 0 0 1 0 0 0

Op: 0 1 0 2 1 1 2 2 2 0 1 0 1 0 1 1

Case	→	*Suggestion*	*Opponent*	*Value*
0001	→	2	0	lost
0001	→	2	1	win
0100	→	1	1	tie
	...			
2221	→	2	1	win
2222	→	0	0	tie

Evolutionary cycle



Selection

- The better individuals must be selected but the worse ones must be given a chance to choose. (stochastic meth.)
  - **Roulette wheel** : the probability of selecting an individual is proportional to its fitness function value
  - **Ranking**: the probability of selecting an individual is proportional to its ranking based on the fitness function values
  - **Tournament**: the best individual is selected from each randomly selected group of the population
  - **Culling**: all individuals below a given threshold are discarded

Recombination

- The code of an offspring must be created from the codes of a pair of parents so that the offspring inherit the properties of their parents.
  - **Crossover**: signals of the parent codes are exchanged at the positions chosen randomly
  - **Recombination**: the corresponding signals of the parent codes are combined

If the code has got some extra invariant condition (permutation), it must be preserved.

Crossover

- **Single or multiple pointed crossover**
  - Some crossover points are chosen at random and in every second section the signals are exchanged.

0	0	1	0	1
1	1	1	1	0

→

0	1	1	0	1
1	0	1	1	0

- **Uniform crossover**
  - Some positions are chosen at random and only the signals on these positions are exchanged.

0	0	1	0	1
1	1	1	1	0

→

1	0	1	0	1
0	1	1	1	0

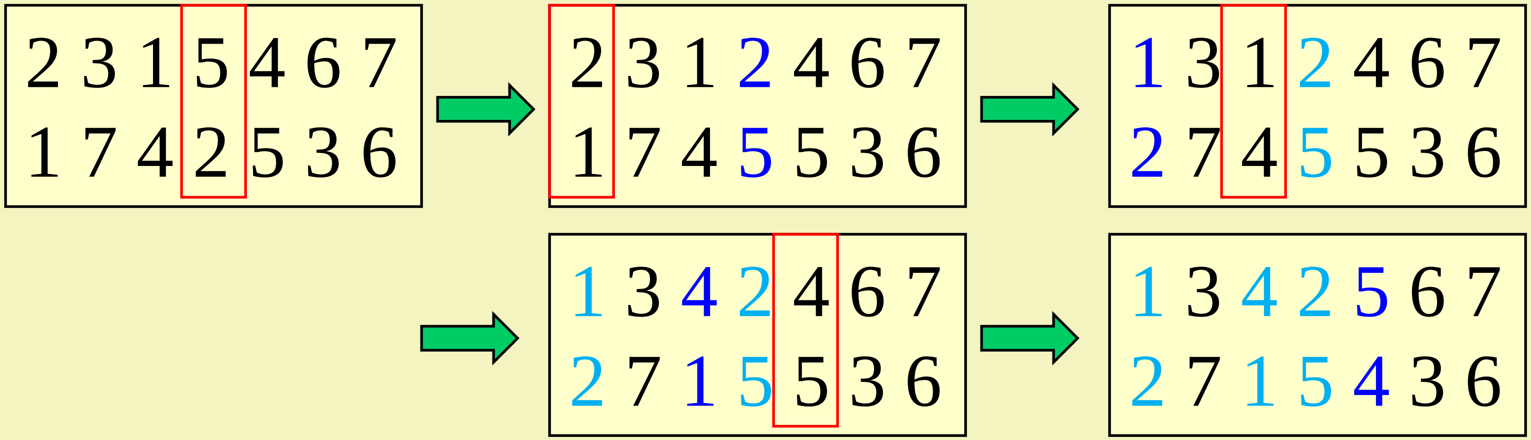
Crossover for permutations 1

- **Partial crossover**
  - Signals are exchanged in one section and then the duplicated signals of the first and the second parent are coupled and exchanged.

2 | 3 1 5 | 4 6 7	→ 2 7 4 2 4 6 7	→ 1 7 4 2 5 6 3
1 | 7 4 2 | 5 3 6	1 3 1 5 5 3 6	2 3 1 5 4 7 6

Crossover for permutations 2

- **Cycle crossover**
  1. Select $i \in [1..length]$ at random
  2. Change $x_i \leftrightarrow y_i$
  3. Find $j \in [1..length]$ $(j \neq i)$ where $x_j = x_i$,
  4. If it is not found then over else $i := j$
  5. goto 2.



Recombinations of real vectors

- **General recombination**
  - The offspring ($\underline{u}$) is somewhere around or in the hyper cuboid spanned by the parents ($\underline{x}$, $\underline{y}$)
  - $\forall i=1 \ldots n: \ u_i = c_i \cdot x_i + (1-c_i) \cdot y_i$

    where $c_i \in [-h, 1+h]$ selected randomly
- **Linear recombination**
  - The offspring ($\underline{u}$) is somewhere on the line matching the parents ($\underline{x}$, $\underline{y}$)
  - $\forall i=1 \ldots n: \ u_i = c \cdot x_i + (1-c) \cdot y_i$

    where $c \in [-h, 1+h]$ selected randomly

Mutation

- Each position of the code is subject to random change with a small independent probability (p).
- In case of a vector:
  - $\forall i=1\ldots n : z_i = x_i \pm range_i \cdot(1-2\cdot p)$
- In case of a binary code:
  - $\forall i=1\ldots n : z_i = 1 - x_i$
- In case of a permutation:
  - swapping signal-pairs
  - shifting the signals in a section cyclically
  - reversing or reordering the signals in a section

Replacement

- Creation of the new population from the elder one and from the new offspring: the selected individuals of the population are substituted with selected offspring.

$$\text{offspring rate}(u) = \frac{\text{number of the successors to be put in}}{\text{size of the population}}$$

$$\text{replacement rate}(v) = \frac{\text{number of the individuals to be removed}}{\text{size of the population}}$$

  - if u=v then total exchange
  - if u<v then some successors must be placed several times
  - if u>v then additional selection is needed of the successors