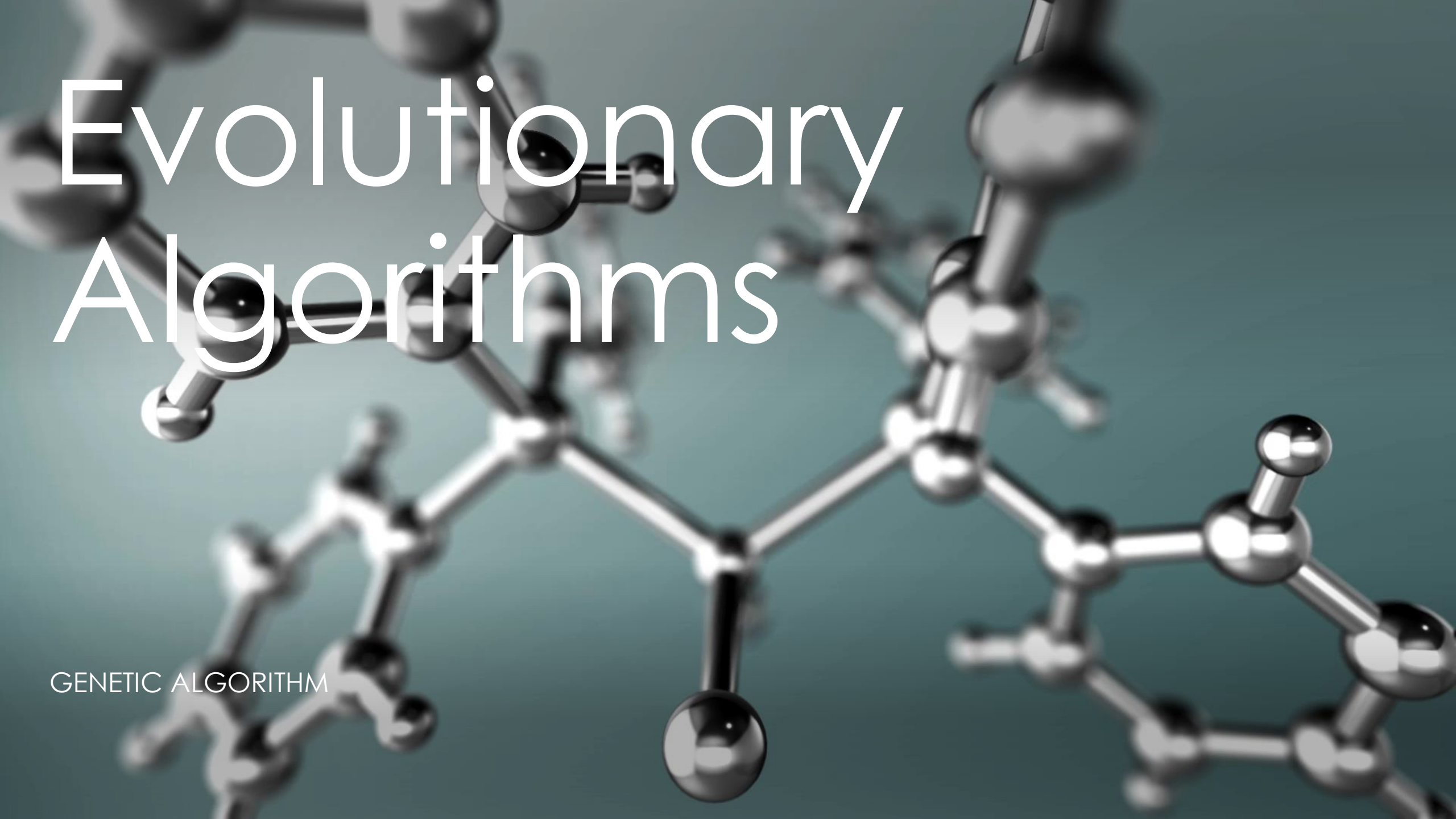


Evolutionary Algorithms

GENETIC ALGORITHM

Evolutionary Algorithms

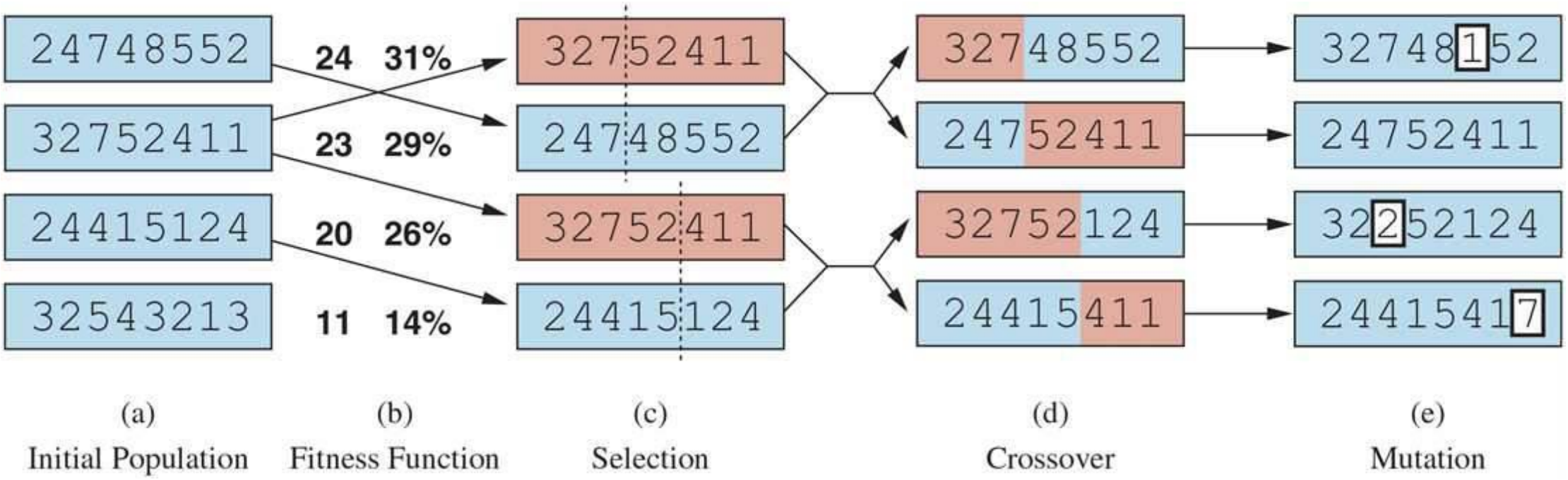
- Variants of Stochastic beam search.
- Explicitly motivated by the metaphor of natural selection in biology!

- **Recombination:** there is a population of individuals (states), in which the fittest (highest value) individuals produce offspring (successor states) that populate the next generation.

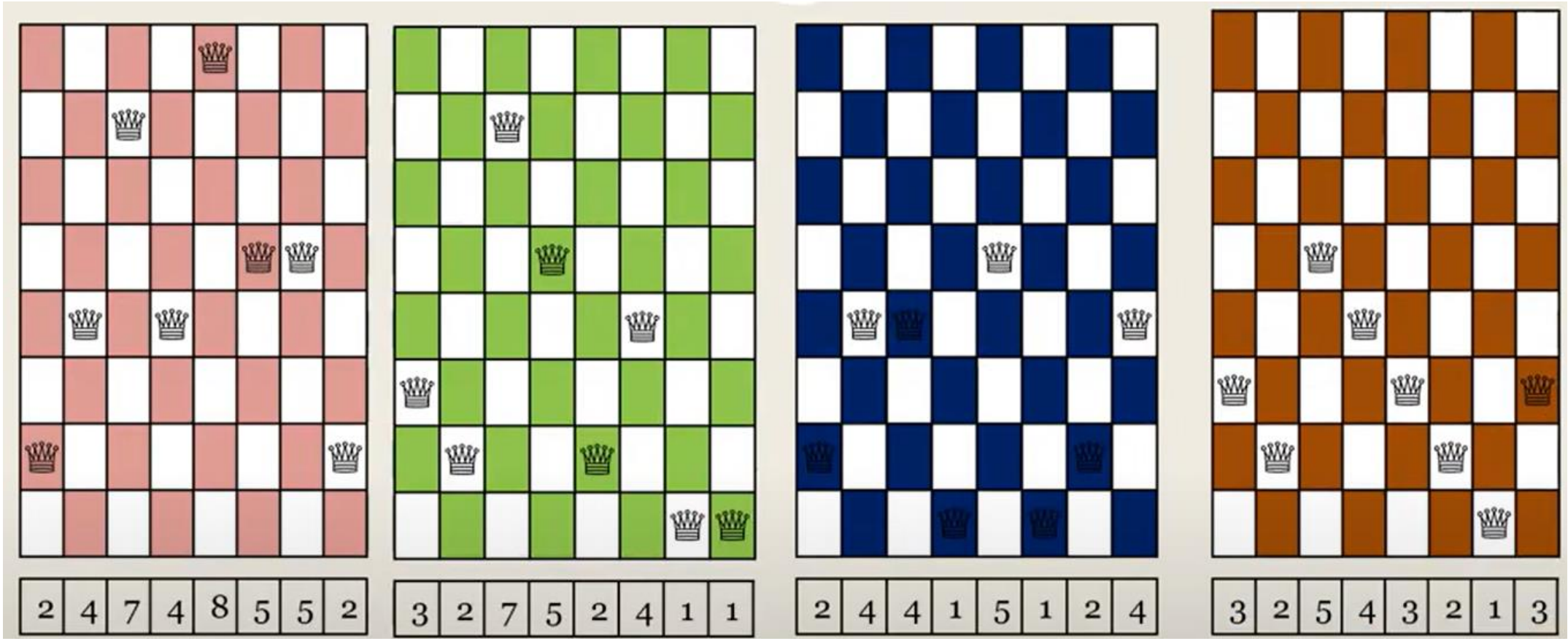
Variants of Evolutionary Algorithms

- Size of the population
- Representation of each individual
- The mixing number ρ
- Selection process ($n > \rho$)
- Recombination procedure
- Mutation rate
- The makeup of next generation

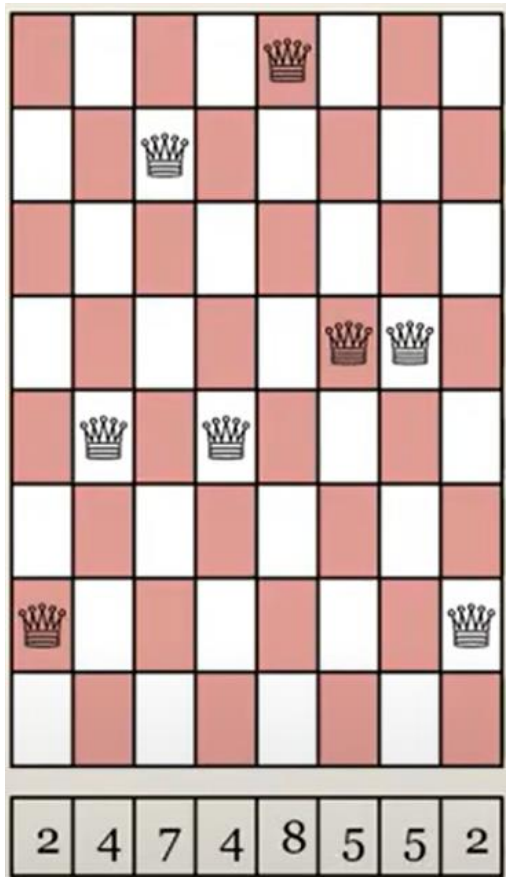
Genetic Algorithm | 8-queens problem



Initial Population

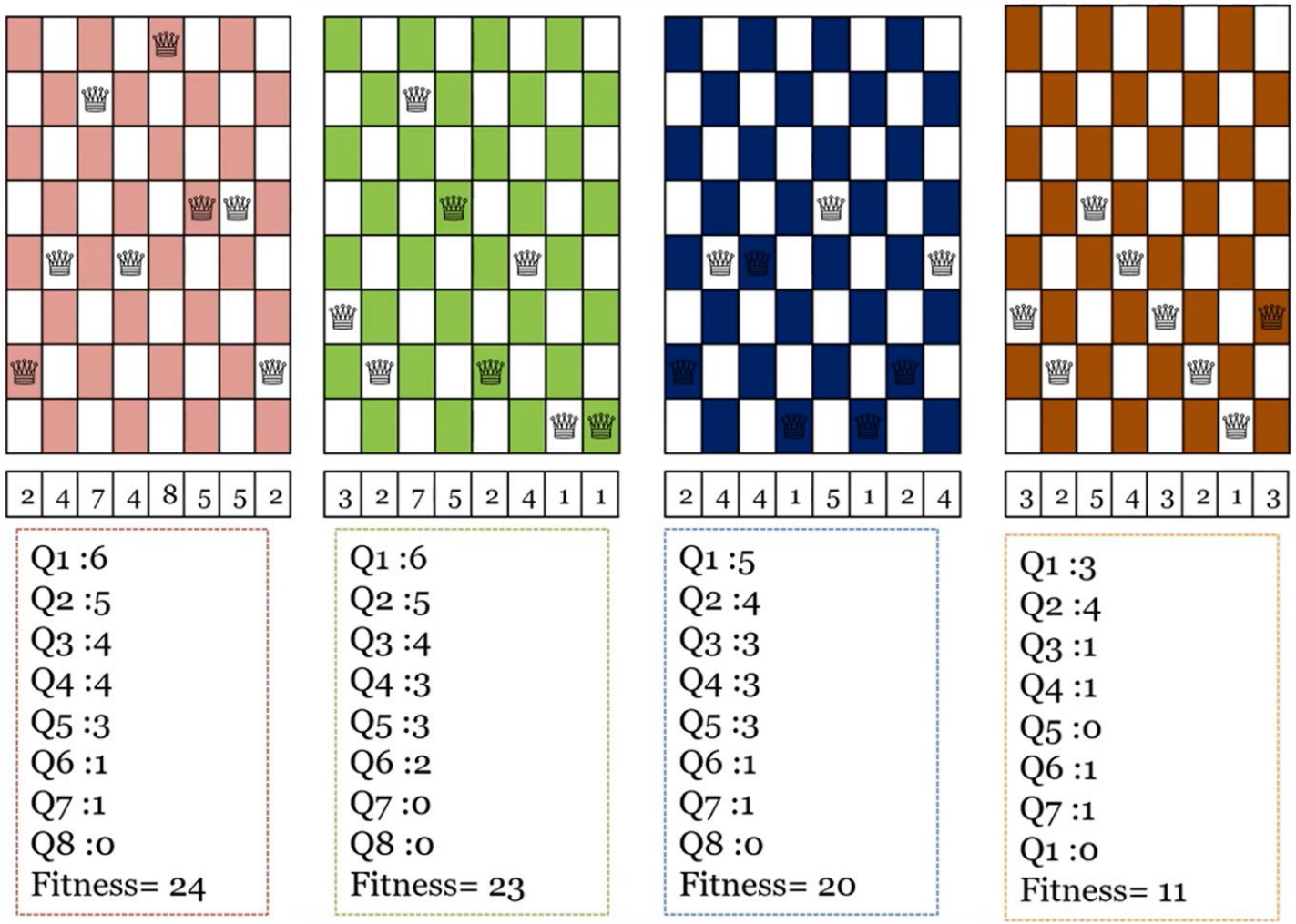


Fitness Function (#Non-Attacking Pairs)



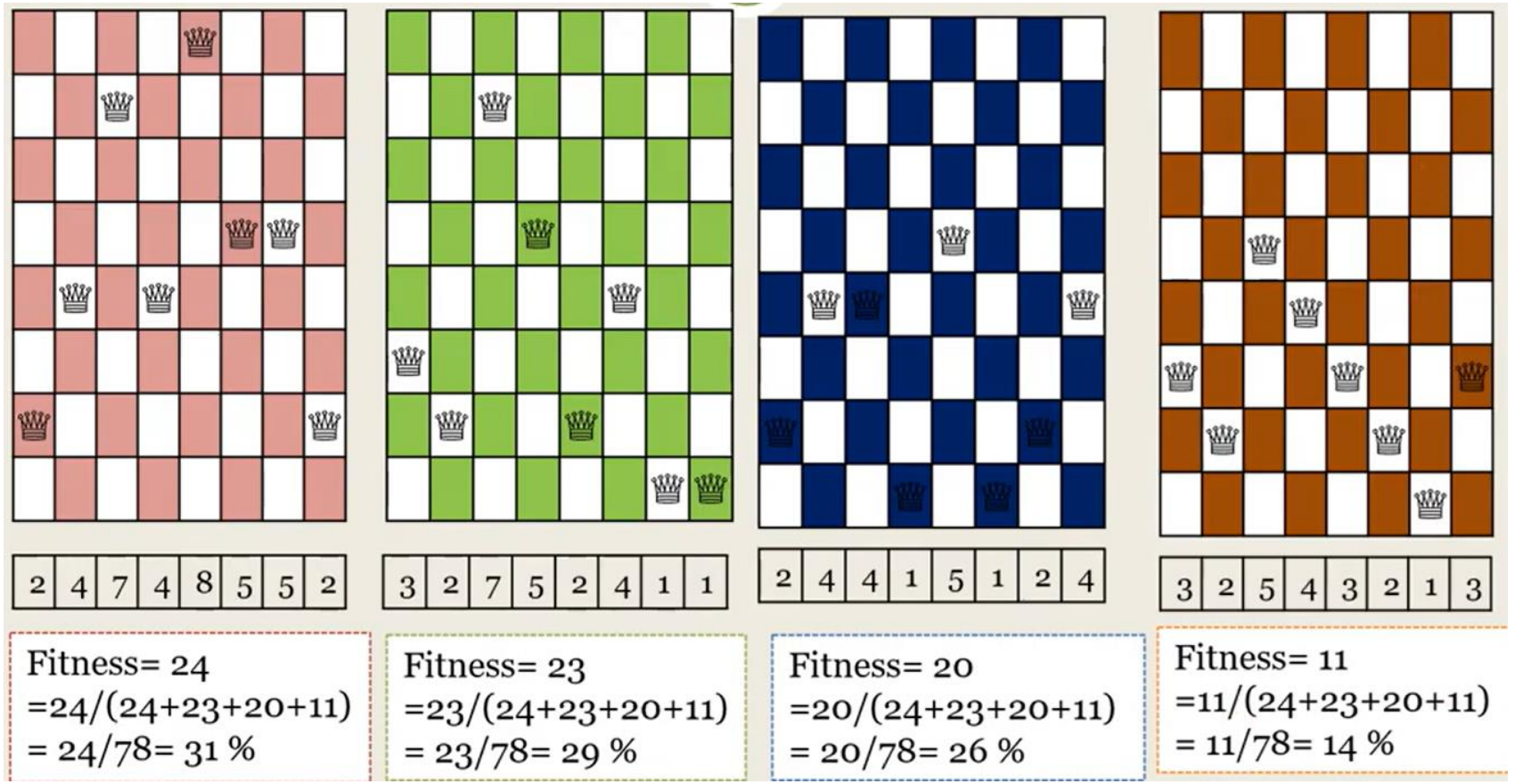
Q1: 6
Q2: 5
Q3: 4
Q4: 4
Q5: 3
Q6: 1
Q7: 1
Q8: 0
Fitness value: 24

Fitness Function (#Non-Attacking Pairs)

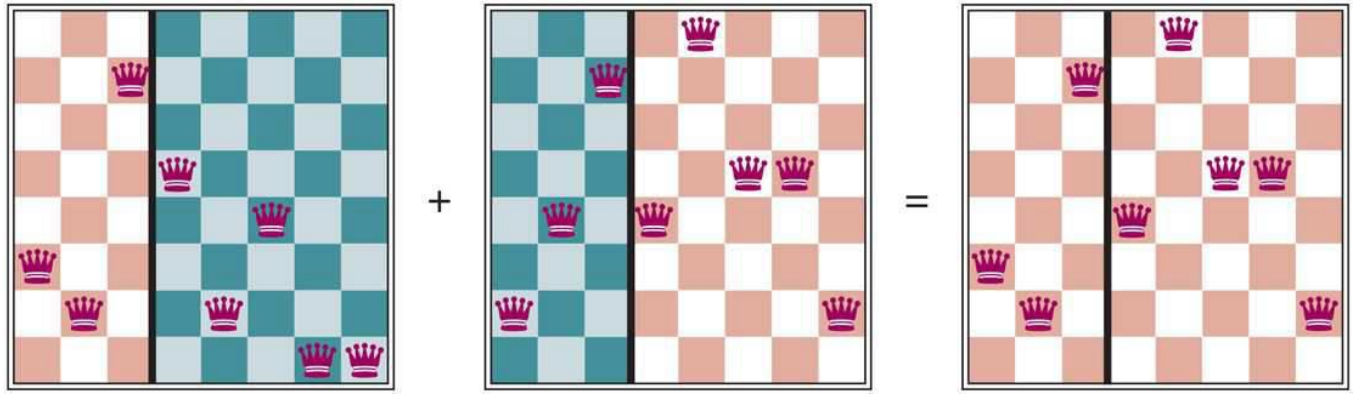


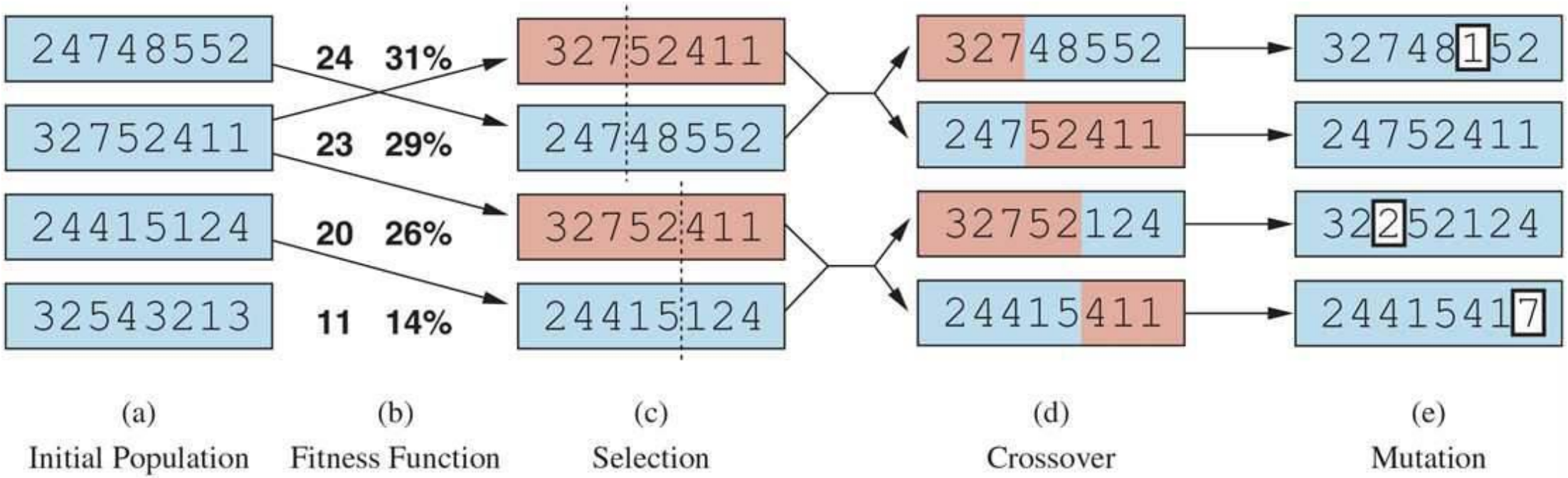
Max value of fitness function?
28

Selection based on Fitness Value

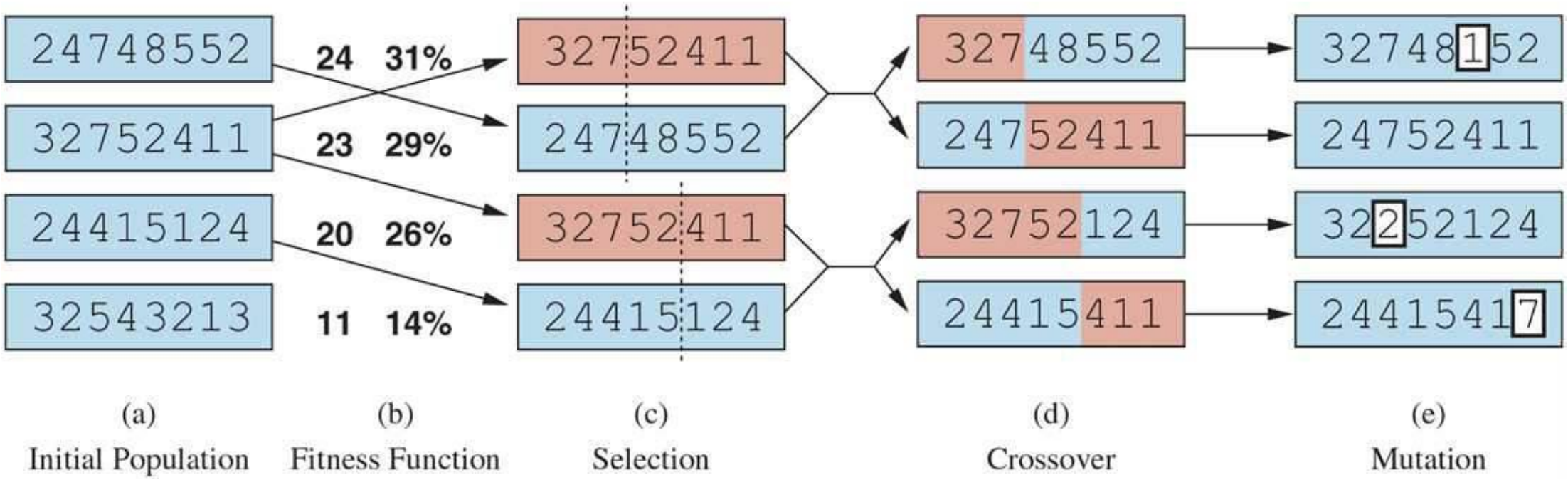


Crossover




24748552
32752411
24415124
32543213
24 31%
23 29%
20 26%
11 14%
32752411
24748552
32752411
24415124
32748552
24752411
32752124
24415411
32748152
24752411
32252124
24415417
(a)
Initial Population
(b)
Fitness Function
(c)
Selection
(d)
Crossover
(e)
Mutation

Mutation


24748552
32752411
24415124
32543213
24 31%
23 29%
20 26%
11 14%
32752411
24748552
32752411
24415124
32748552
24752411
32752124
24415411
32748152
24752411
32252124
24415417
(a)
Initial Population
(b)
Fitness Function
(c)
Selection
(d)
Crossover
(e)
Mutation

Genetic Algorithm

- https://algorithmsinreallife.wordpress.com/2020/05/15/genetic-algorithm/

```
function GENETIC-ALGORITHM(population, fitness) returns an individual
  repeat
    weights ← WEIGHTED-BY(population, fitness)
    population2 ← empty list
    for i = 1 to SIZE(population) do
      parent1, parent2 ← WEIGHTED-RANDOM-CHOICES(population, weights, 2)
      child ← REPRODUCE(parent1, parent2)
      if (small random probability) then child ← MUTATE(child)
      add child to population2
    population ← population2
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to fitness

function REPRODUCE(parent1, parent2) returns an individual
  n ← LENGTH(parent1)
  c ← random number from 1 to n
  return APPEND(SUBSTRING(parent1, 1, c), SUBSTRING(parent2, c + 1, n))
```