Genetic Algorithms

Genetic Algorithms

- Genetic algorithms provide an approach to learning that is based loosely on simulated evolution.
- Hypotheses are often described by bit strings whose interpretation depends on the application.
- The search for an appropriate hypothesis begins with a population of initial hypotheses.
- Members of the current population give rise to the next generation population by means of operations such as random mutation and crossover, which are patterned after processes in biological evolution.
- The hypotheses in the current population are evaluated relative to a given measure of fitness, with the most fit hypotheses selected probabilistically as seeds for producing the next generation.

Genetic Algorithms

- Genetic algorithms (GAs) provide a learning method motivated by an analogy to biological evolution.
- GAs generate successor hypotheses by repeatedly mutating and recombining parts of the best currently known hypotheses.
- At each step, the current population is updated by replacing some fraction of the population by offspring of the most fit current hypotheses.
- The process forms a generate-and-test beam-search of hypotheses, in which variants of the best current hypotheses are most likely to be considered next.

Genetic Algorithms

- GAs search a space of candidate hypotheses to identify the best hypothesis.
- In GAs the "best hypothesis" is defined as the one that optimizes a predefined numerical measure for the problem at hand, called the hypothesis ***fitness.***
- For example, if the learning task is the problem of approximating an unknown function given training examples of its input and output, then fitness could be defined as the accuracy of the hypothesis over this training data.

A prototypical genetic algorithm

GA($Fitness$, $Fitness\_threshold$, p, r, m)

Fitness: A function that assigns an evaluation score, given a hypothesis.

$Fitness\_threshold$: A threshold specifying the termination criterion.

p: The number of hypotheses to be included in the population.

r: The fraction of the population to be replaced by Crossover at each step.

m: The mutation rate.

- *Initialize population:* $P \leftarrow$ Generate p hypotheses at random
- *Evaluate:* For each h in P, compute $Fitness(h)$

A prototypical genetic algorithm

- While $[\max_h Fitness(h)] < Fitness\_threshold$ do

  Create a new generation, P_S:

  1. *Select:* Probabilistically select $(1-r)p$ members of P to add to P_S. The probability $\Pr(h_i)$ of selecting hypothesis h_i from P is given by

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

  2. *Crossover:* Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P, according to $\Pr(h_i)$ given above. For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by applying the Crossover operator. Add all offspring to P_s.
  3. *Mutate:* Choose m percent of the members of P_s with uniform probability. For each, invert one randomly selected bit in its representation.
  4. *Update:* $P \leftarrow P_s$.
  5. *Evaluate:* for each h in P, compute $Fitness(h)$

- Return the hypothesis from P that has the highest fitness.

Selection of Hypotheses

- A certain number of hypotheses from the current population are selected for inclusion in the next generation.
- These are selected ***probabilistically,*** where the probability of selecting hypothesis h_i is given by

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

- The probability that a hypothesis will be selected is proportional to its own fitness and is inversely proportional to the fitness of the other competing hypotheses in the current population.

Representing Hypotheses

- Hypotheses in GAs are often represented by bit strings, so that they can be easily manipulated by genetic operators such as mutation and crossover.
- How if-then rules can be encoded by bit strings
  - Consider the attribute ***Outlook,*** which can take on any of the three values ***Sunny, Overcast,*** or ***Rain.***
  - The string 010 represents the constraint that ***Outlook*** must take on the second of these values, or ***Outlook*** = ***Overcast.***
  - The string 011 represents the more general constraint that allows two possible values, or ***(Outlook*** = ***Overcast or Rain).***
  - The string 111 represents the most general possible constraint, indicating that we don't care which of its possible values the attribute takes on.

Representing Hypotheses

- The following rule precondition can be represented by the following bit string of length five.

$$(Outlook = Overcast \vee Rain) \wedge (Wind = Strong) \rightarrow$$

Outlook	*Wind*
011	10

- An entire rule can be described by concatenating the bit strings describing the rule preconditions, together with the bit string describing the rule postcondition.

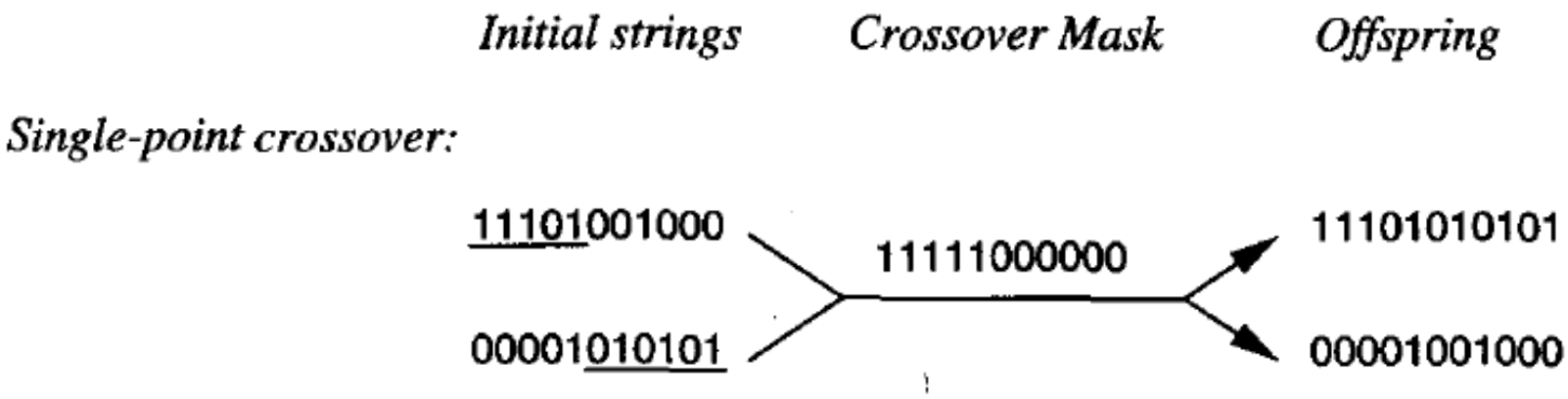
IF $Wind = Strong$ THEN $PlayTennis = yes$

Outlook	*Wind*	*PlayTennis*
111	10	10

Crossover Operator

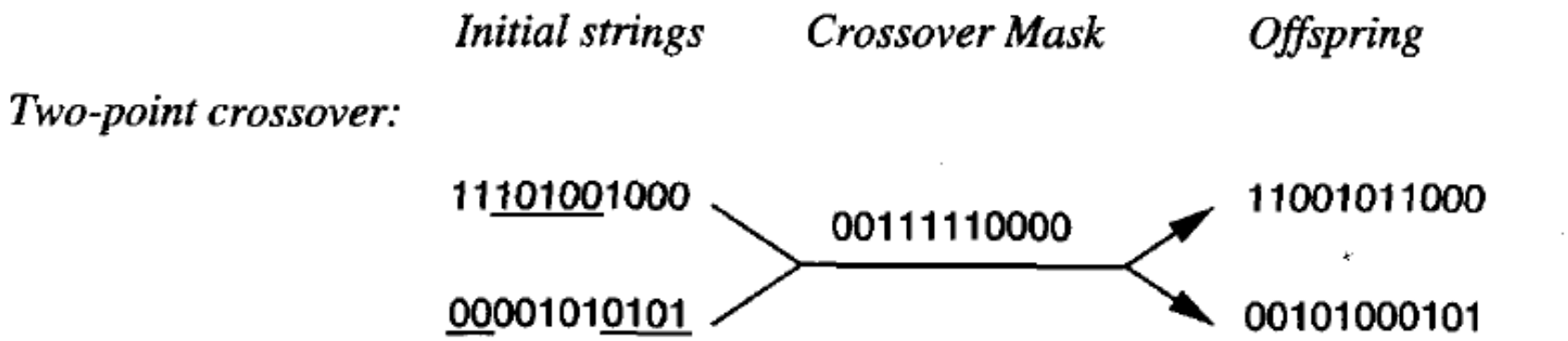
- The ***crossover operator*** produces two new offspring from two parent strings, by copying selected bits from each parent.
- The bit at position i in each offspring is copied from the bit at position i in one of the two parents.
- The choice of which parent contributes the bit for position i is determined by an additional string called the ***crossover mask.***
- There are different crossover operators.
  - ***Single-point crossover***
  - ***Two-point crossover***
  - ***Uniform crossover***

Single-point crossover



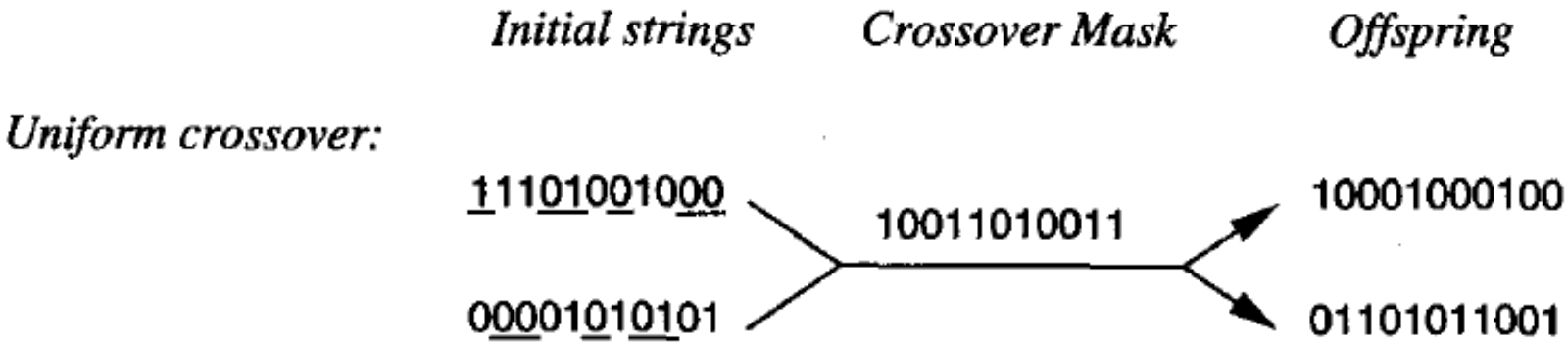
- In single-point crossover, the crossover mask is always constructed so that it begins with a string containing ***n*** contiguous 1s, followed by the necessary number of 0s to complete the string.
- This results in offspring in which the first n bits are contributed by one parent and the remaining bits by the second parent.
- Each time the single-point crossover operator is applied, the crossover point ***n*** is chosen at random, and the crossover mask is then created and applied.

Two-point crossover



- In ***two-point crossover,*** offspring are created by substituting intermediate segments of one parent into the middle of the second parent string.
- The crossover mask is a string beginning with **no** zeros, followed by a contiguous string of **nl** ones, followed by the necessary number of zeros to complete the string.
- Each time the two-point crossover operator is applied, a mask is generated by randomly choosing the integers **no** and **nl.**
- Two offspring are created by switching the roles played by the two parents.

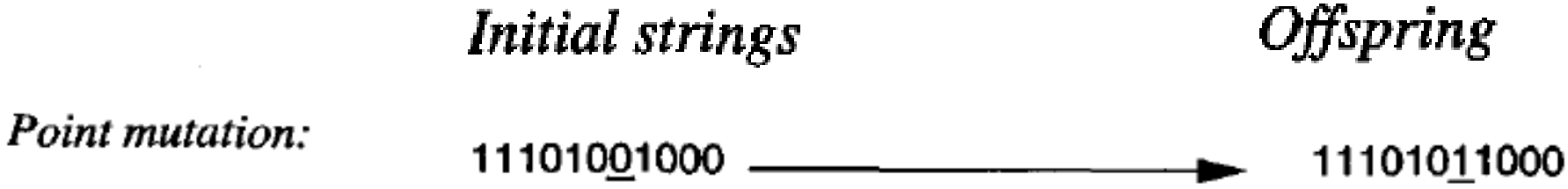
Uniform crossover



- ***Uniform crossover*** combines bits sampled uniformly from the two parents.
- The crossover mask is generated as a random bit string with each bit chosen at random and independent of the others.

Mutation

- ***Mutation*** operator produces offspring from a single parent.
- The ***mutation*** operator produces small random changes to the bit string by choosing a single bit at random, then changing its value.
- Mutation is often performed after crossover has been applied.



Fitness Function

- The fitness function defines the criterion for ranking potential hypotheses and for probabilistically selecting them for inclusion in the next generation population.
- If the task is to learn classification rules, then the fitness function typically has a component that scores the classification accuracy of the rule over a set of provided training examples.
- Often other criteria may be included as well, such as the complexity or generality of the rule.
- More generally, when the bit-string hypothesis is interpreted as a complex procedure (e.g., when the bit string represents a collection of if-then rules), the fitness function may measure the overall performance of the resulting procedure rather than performance of individual rules.

Fitness Function and Selection

- The probability that a hypothesis will be selected is given by the ratio of its fitness to the fitness of other members of the current population.
- This method is called ***fitness proportionate selection,*** or ***roulette wheel selection***

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

tournament selection

- two hypotheses are first chosen at random from the current population.
- With some predefined probability p the more fit of these two is then selected, and with probability (1 - p) the less fit hypothesis is selected.

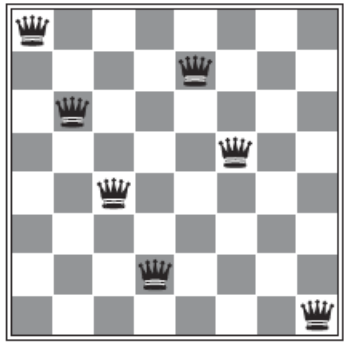
rank selection,

- the hypotheses in the current population are first sorted by fitness.
- The probability that a hypothesis will be selected is then proportional to its rank in this sorted list, rather than its fitness.

Genetic Algorithms

8-queens problem

- In 8-queens problem, we try put 8 queens on a chess board so that none of them attacks another queen.

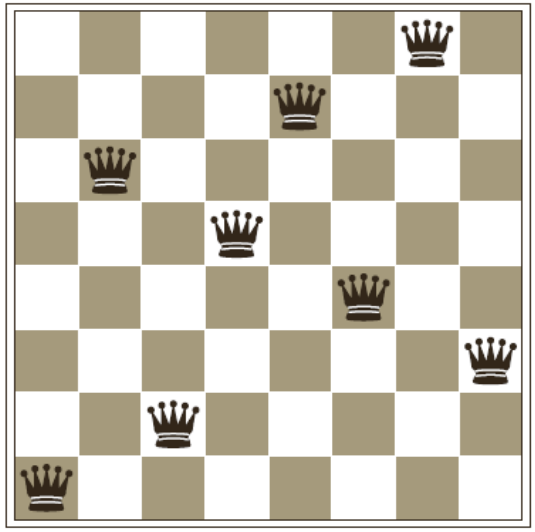


- Each individual is represented as a string over a finite alphabet—most commonly, a string of 0s and 1s.
- An 8-queens state must specify the positions of 8 queens, each in a column of 8 squares, and so it requires $8 \times \log_2 8 = 24$ bits.
- Alternatively, the individual could be represented as 8 digits, each in the range from 1 to 8.

Genetic Algorithms

8-queens problem

- An individual can be represented using a 8 digit string.
- Each digit in the range from 1 to 8 to indicate the position of the queen in that column

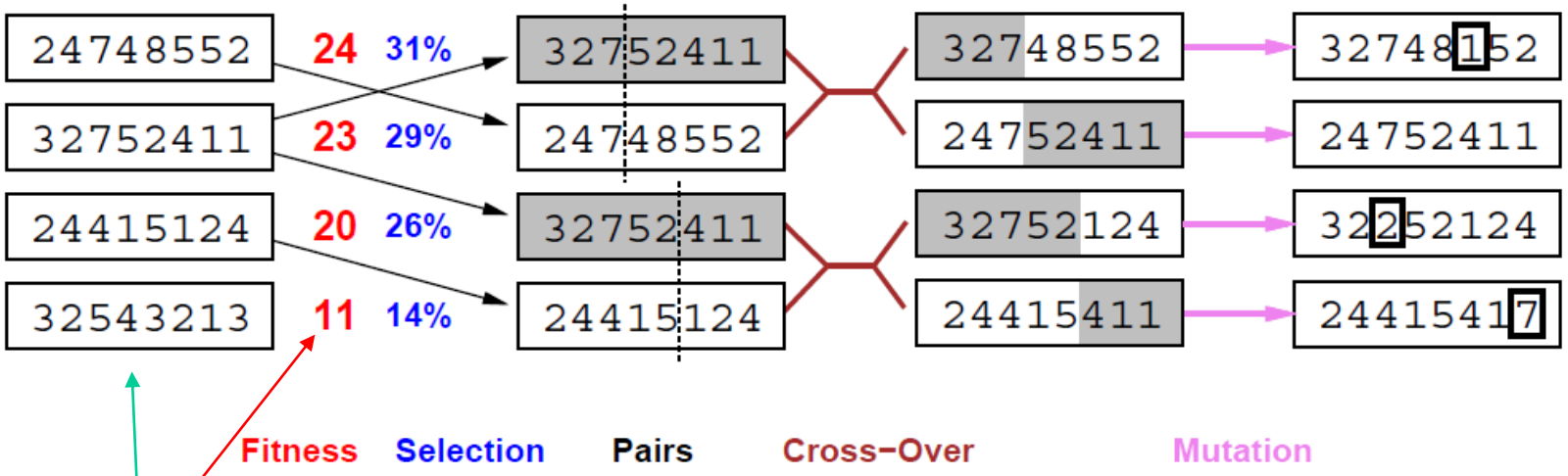


➔

1 6 2 5 7 4 8 3

Genetic Algorithms

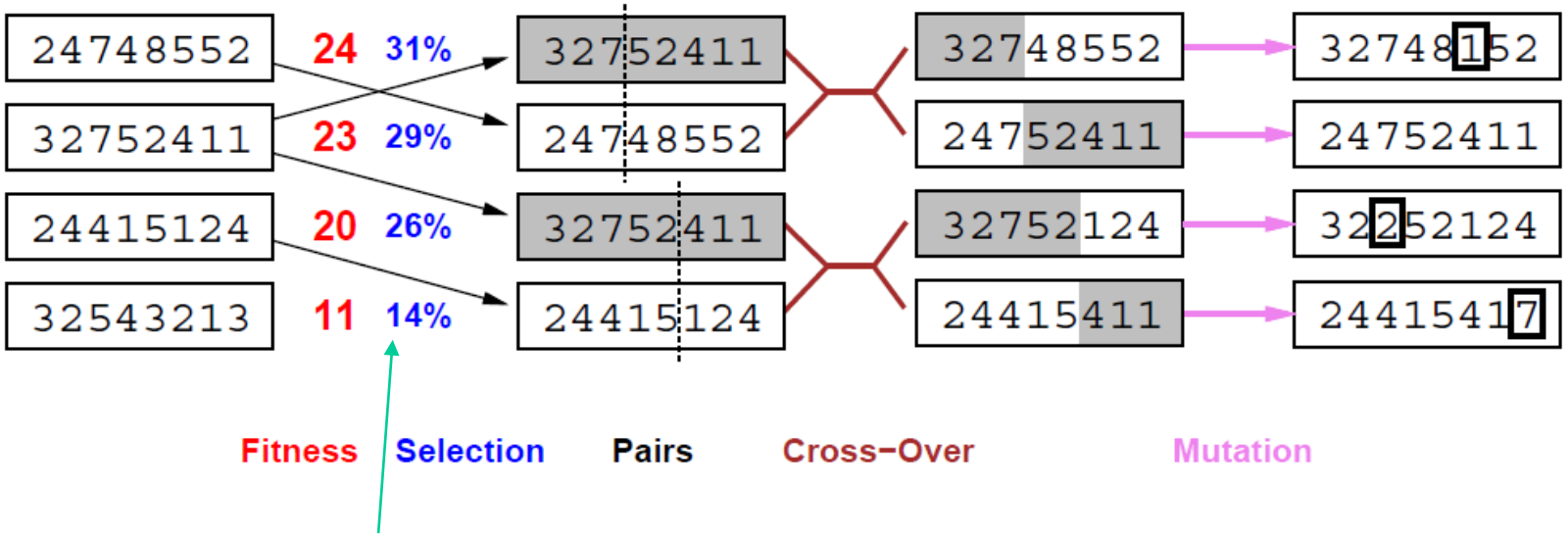
8-queens problem



- The **initial population** has 4 states.
- A **fitness function** should return higher values for better states, so, for the 8-queens problem we use the number of non-attacking pairs of queens, which has a value of 28 for a solution.
  - The values of the four states are 24, 23, 20, and 11.

Genetic Algorithms

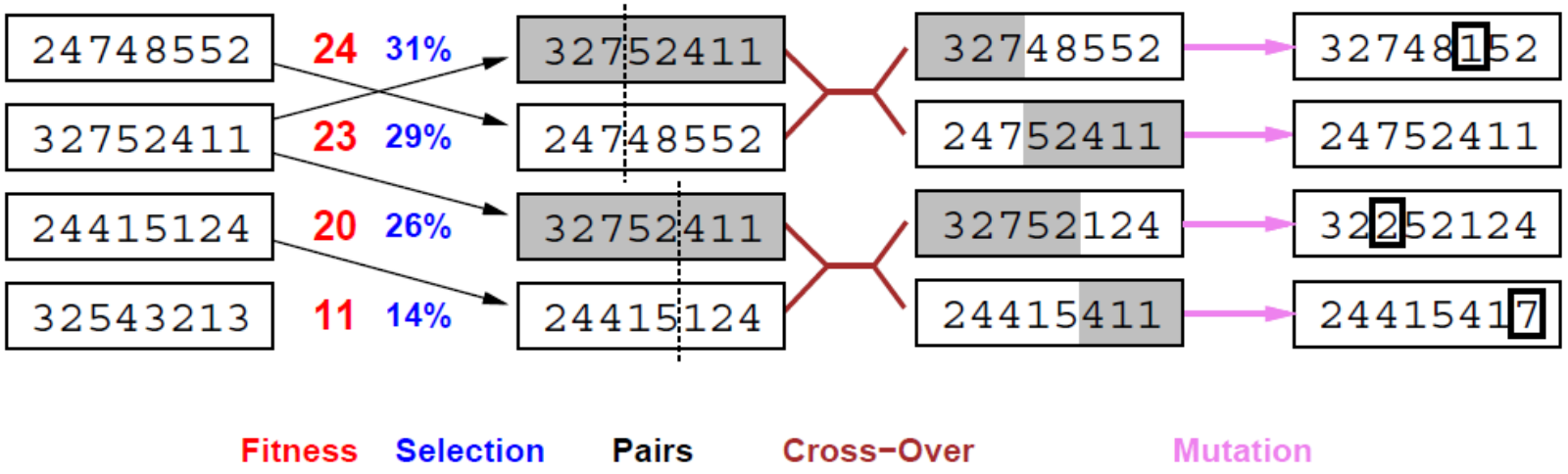
8-queens problem



- The probability of being chosen for reproducing is directly proportional to the fitness score.
- Two pairs are selected at random for reproduction, in accordance with the probabilities.
  - Notice that one individual is selected twice and one not at all.

Genetic Algorithms

8-queens problem

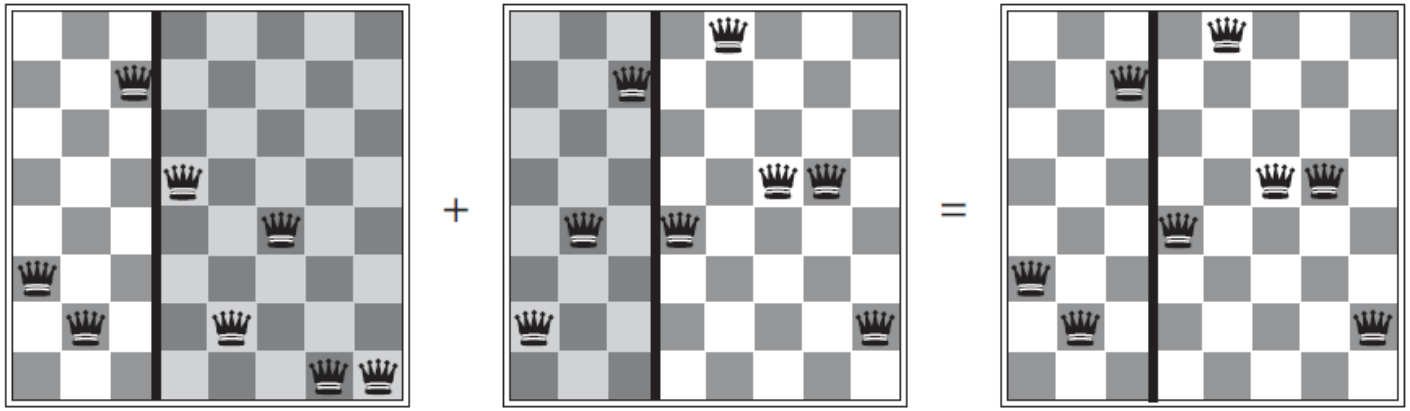


- The **crossover points** are after third digit in first pair and after fifth digit in second pair.
- The first child of the first pair gets the first three digits from the first parent and the remaining digits from the second parent, whereas the second child gets the first three digits from the second parent and the rest from the first parent.

Genetic Algorithms

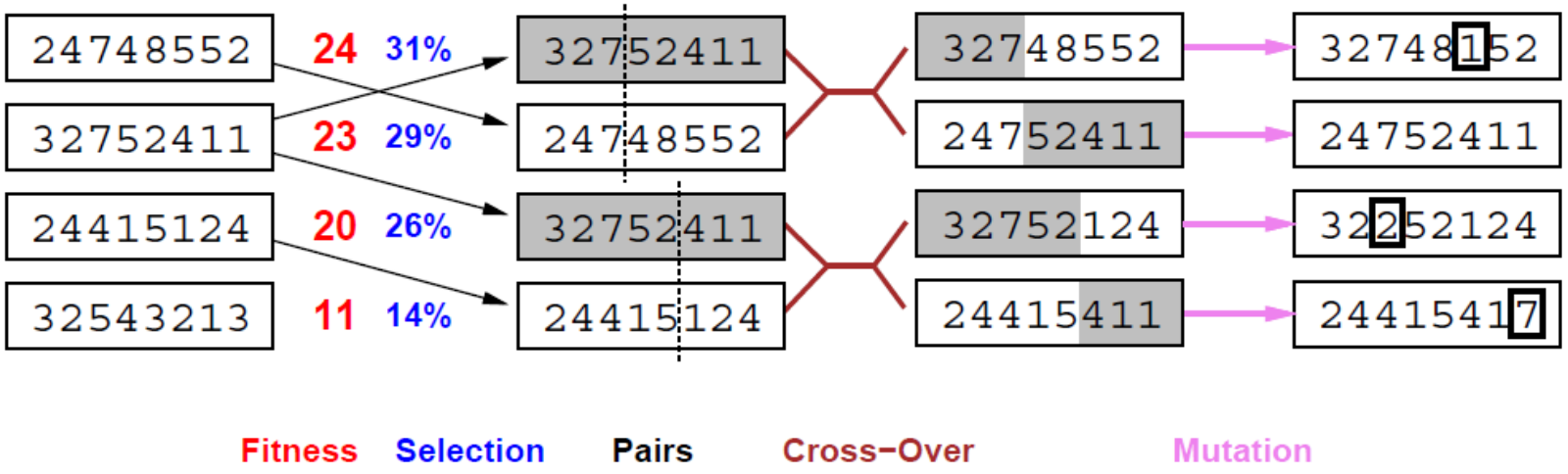
8-queens problem

- Crossover helps iff substrings are meaningful components.
- The shaded columns are lost in the crossover step and the unshaded columns are retained.



Genetic Algorithms

8-queens problem



- One digit was **mutated** in the first, third, and fourth offspring.
- In the 8-queens problem, this corresponds to choosing a queen at random and moving it to a random square in its column.