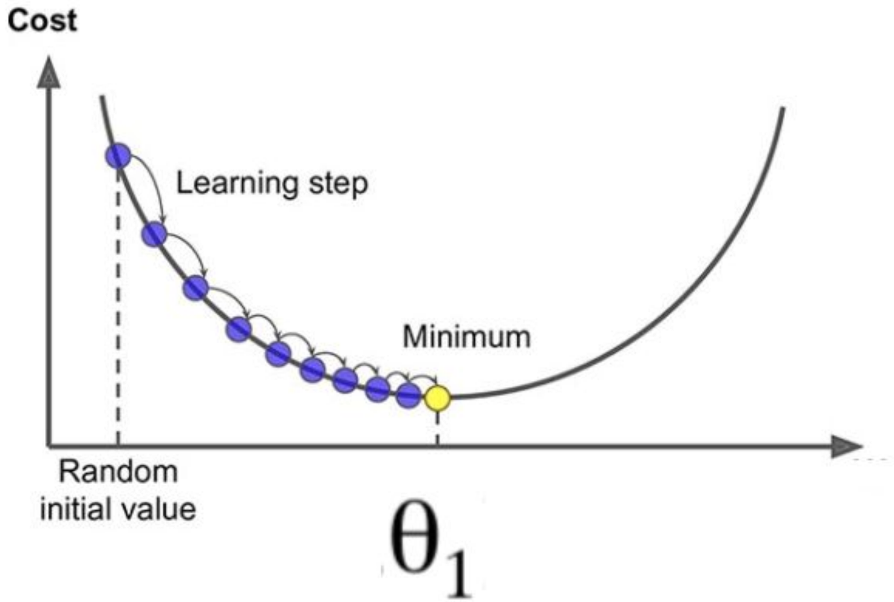
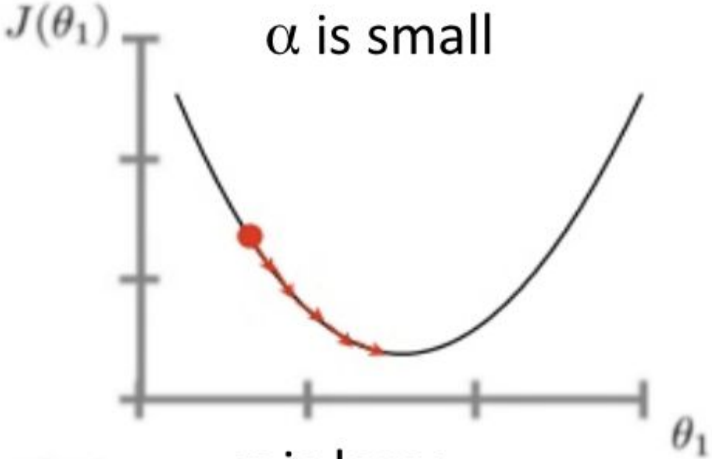
Genetic algorithm

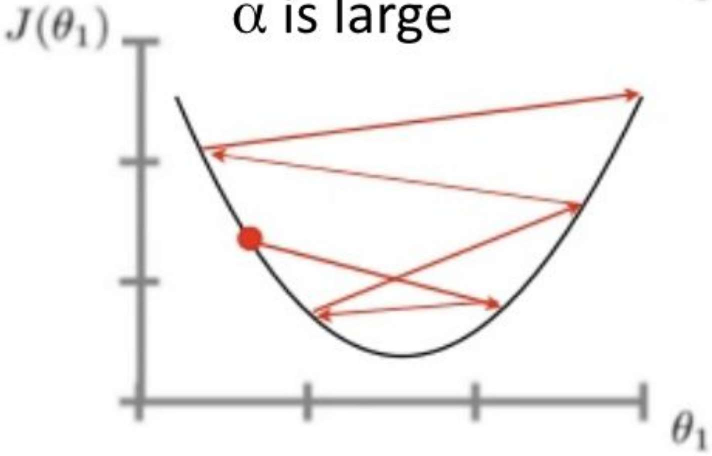
Python for AI

Optimizer: Gradient Descent

```
repeat until convergence {
    $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$
        (for $j = 1$ and $j = 0$)
}
```





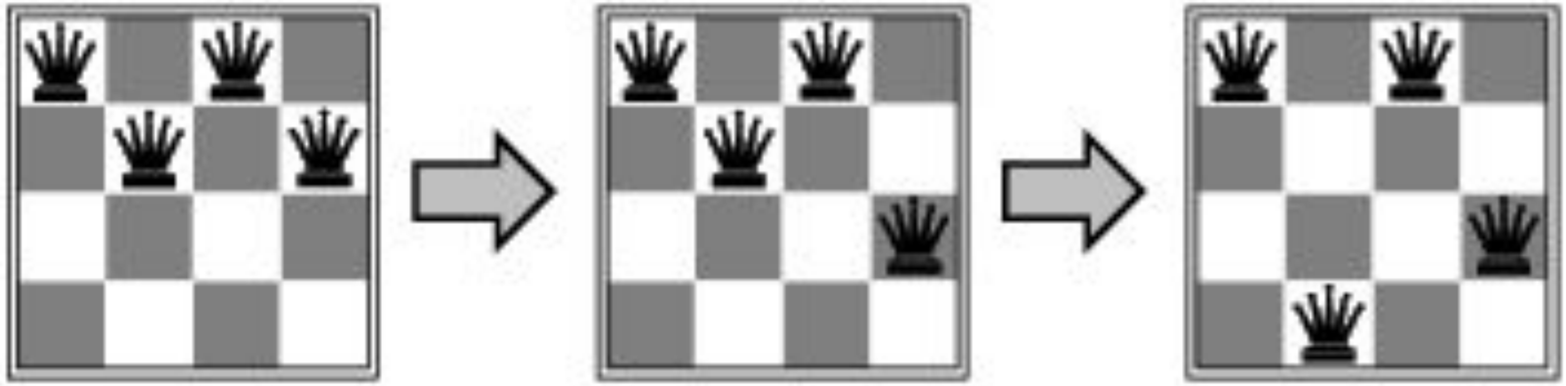


Optimization

- Derivative-based Optimization
  - Descent Methods
  - The Method of Steepest Descent
  - Classical Newton's Method
  - Step Size Determination
- Derivative-free Optimization
  - Genetic Algorithms
  - Simulated Annealing
  - Random Search
  - Downhill Simplex Search

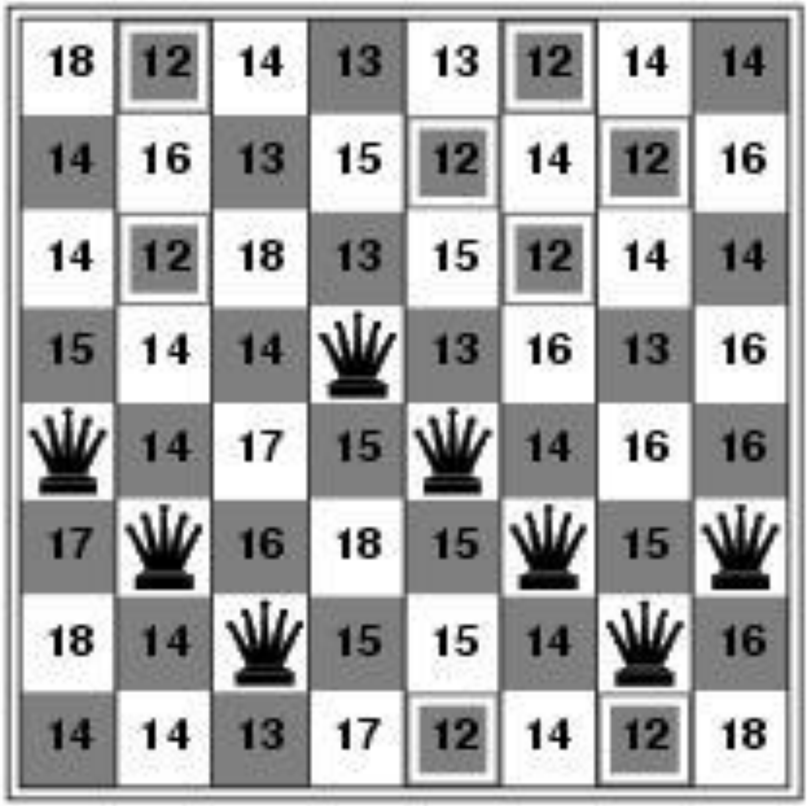
Example: *n*-queens

- Put *n* queens on an *n* x *n* board with no two queens on the same row, column, or diagonal
  - Note different search space… all states have N queens



- Is it a satisfaction problem or optimization?

8-queens problem



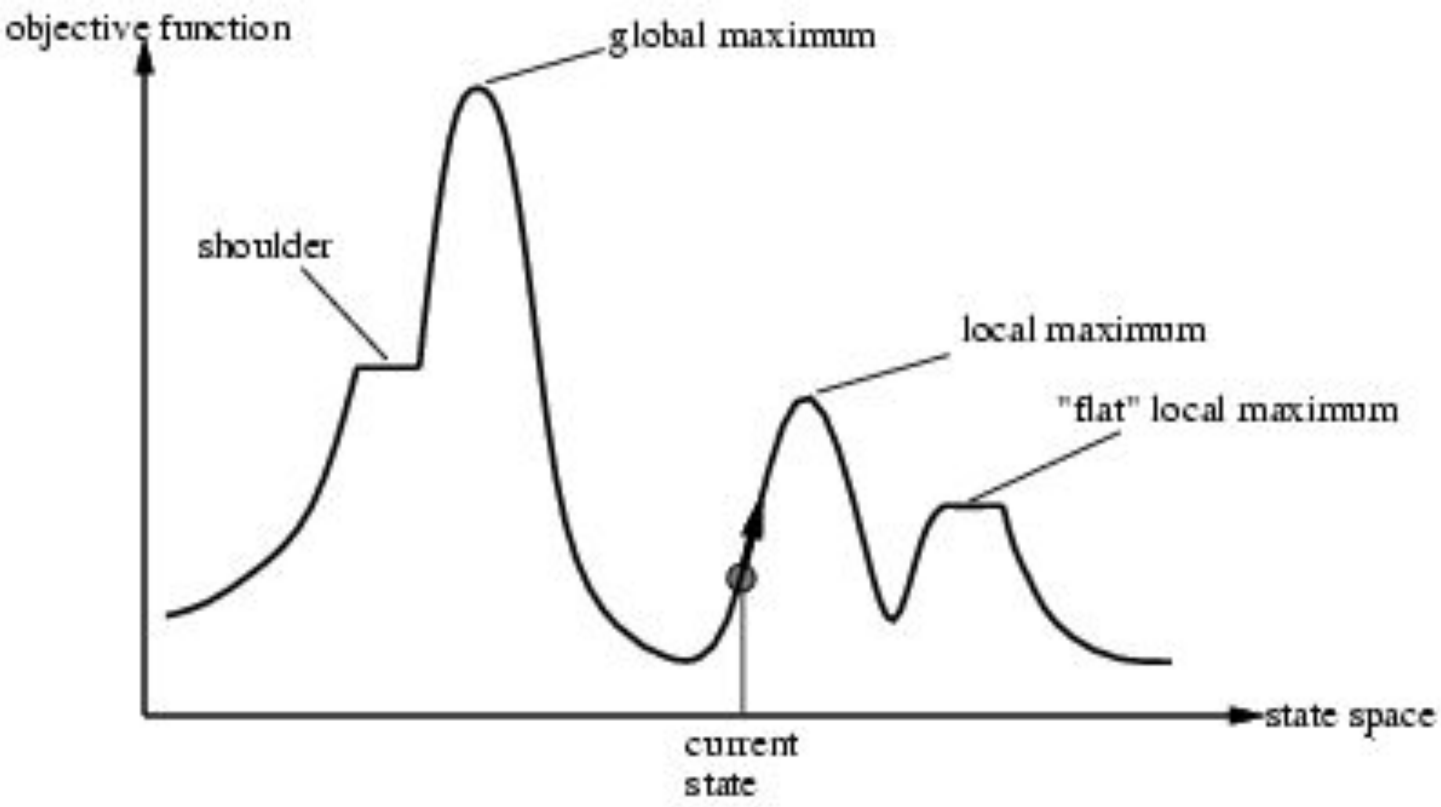
18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♛	13	16	13	16
♛	14	17	15	♛	14	16	16
17	♛	16	18	15	♛	15	♛
18	14	♛	15	15	14	♛	16
14	14	13	17	12	14	12	18

- Need heuristic function
  - Convert to an optimization problem
- *h* = number of ***pairs*** of queens attacking each other
- *h = 17* for the above state

Search Space Recap

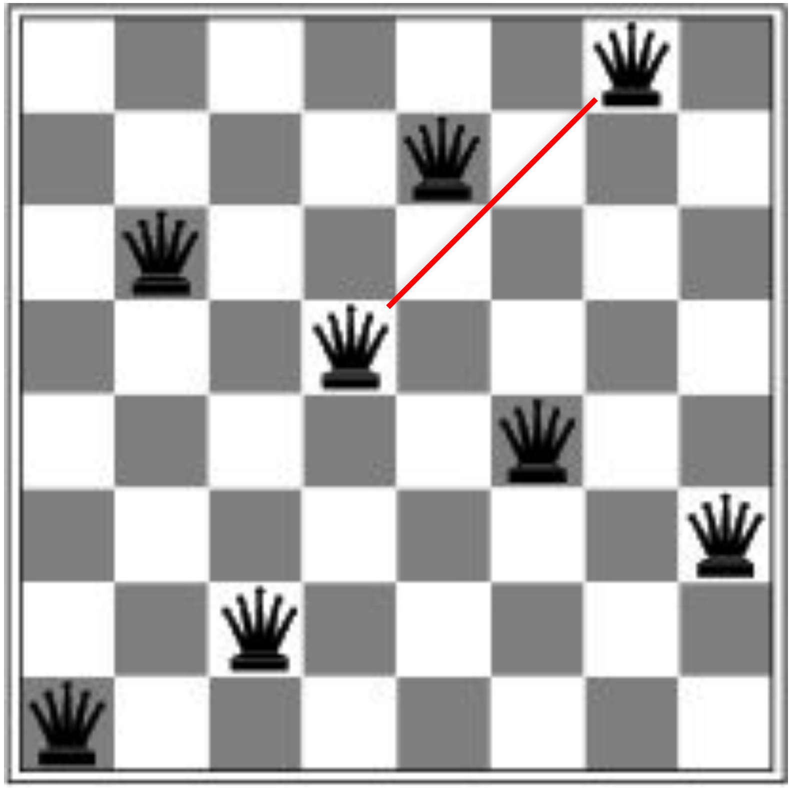
- State
  - All N queens on the board in some configuration
- Successor function
  - Move single queen to another square in same column.
- Example of a heuristic function *h(n)*:
  - the # of queens-pairs that are attacking each other
  - (we want to minimize this)

Hill climbing



Hill Climbing gets stuck in local maxima

Hill-climbing search: 8-queens problem



- Is this a solution?
- What is h?
- Is any successor better?

Hill-climbing on 8-queens

- Randomly generated 8-queens starting states…
- 14% the time it solves the problem
- 86% of the time it get stuck at a local minimum

- However…
  - Takes only 4 steps on average when it succeeds
  - And 3 on average when it gets stuck
  - (for a state space with 8^8 =~17 million states)

Hill-climbing with random restarts

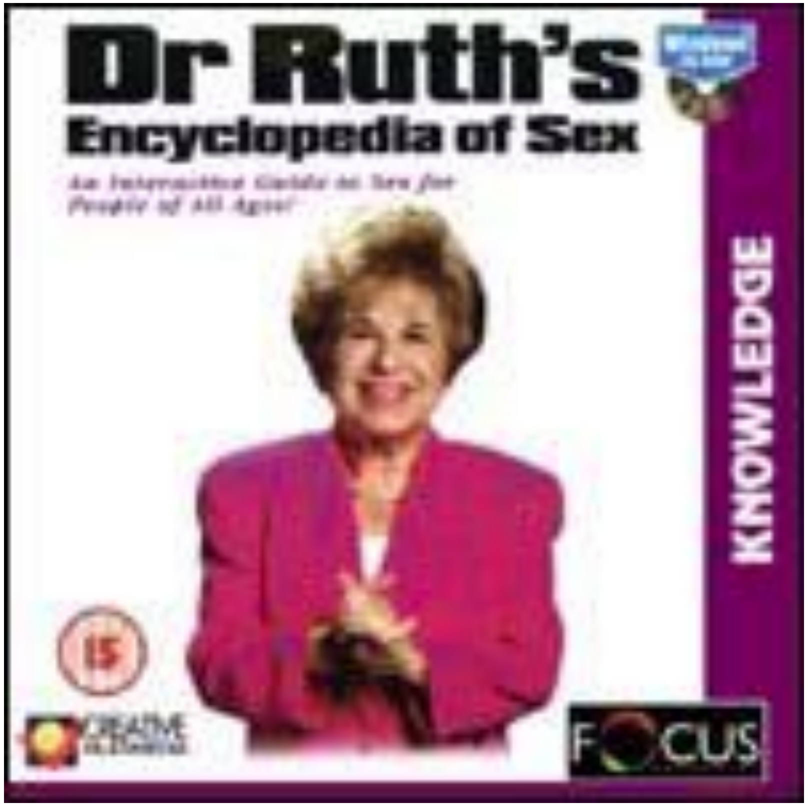
- If at first you don’t succeed, try, try again!
- Different variations
  - For each restart: run until termination vs. run for a fixed time
  - Run a fixed number of restarts or run indefinitely

Use this algorithm!

- Analysis
  - Say each search has probability p of success
    - E.g., for 8-queens, p = 0.14 with no sideways moves
  - Expected number of restarts?

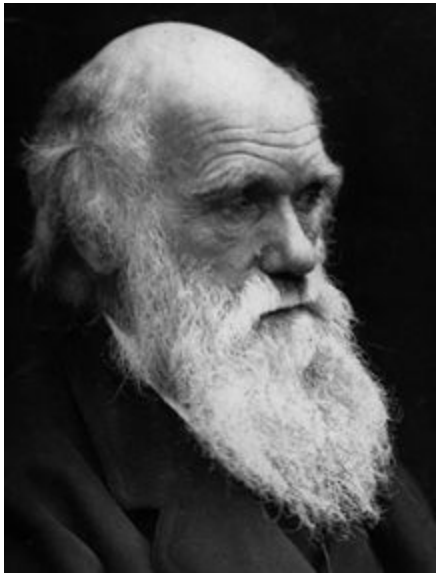
Restarts	0	2	4	8	16	32	64
Success?	14%	36%	53%	74%	92%	99%	99.994%

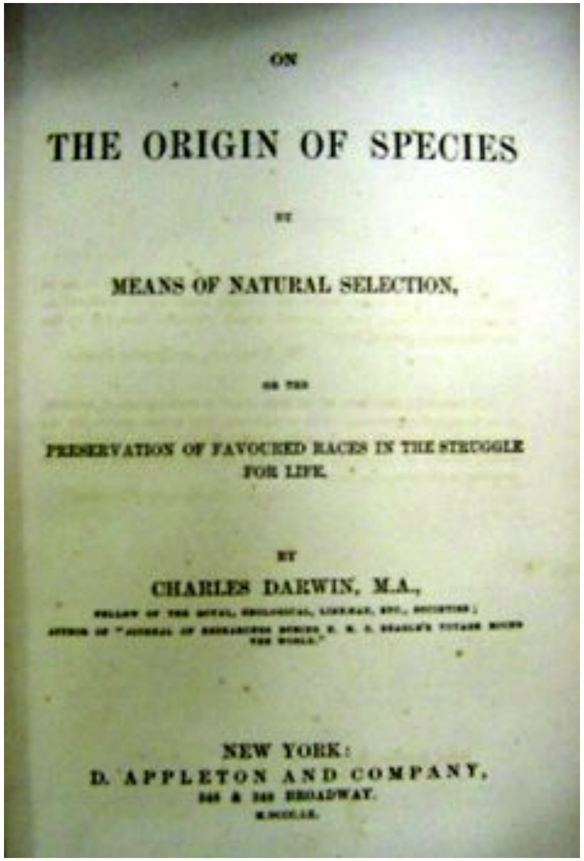
  - Expected number of steps taken?



Hey! Perhaps sex can improve search?

Sure! Check out ye book.





ON

THE ORIGIN OF SPECIES

BY

MEANS OF NATURAL SELECTION,

OR THE

PRESERVATION OF FAVOURED RACES IN THE STRUGGLE FOR LIFE.

BY

CHARLES DARWIN, M.A.,

FELLOW OF THE ROYAL, GEOLOGICAL, LINNÆAN, ETC., SOCIETIES;
AUTHOR OF "JOURNAL OF RESEARCHES DURING H. M. S. BEAGLE'S VOYAGE ROUND THE WORLD."

NEW YORK:
D. APPLETON AND COMPANY,
346 & 348 BROADWAY.
M.DCCC.LX.

Genetic algorithms

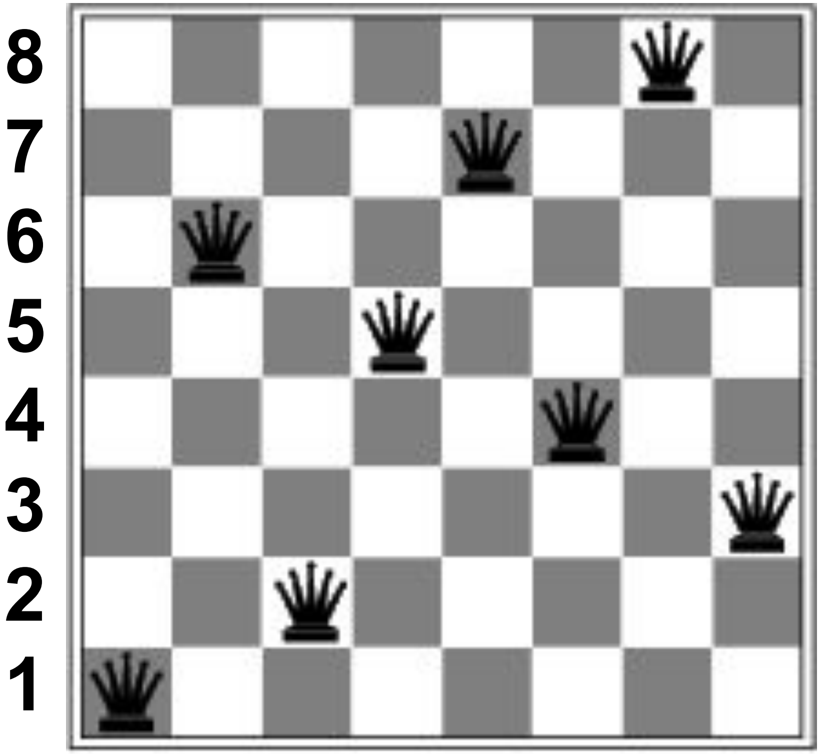
- Twist on Local Search: successor is generated by combining two parent states

- A state is represented as a string over a finite alphabet (e.g. binary)
  - 8-queens
    - State = position of 8 queens each in a column

- Start with *k* randomly generated states (population)

- Evaluation function (fitness function):
  - Higher values for better states.
  - Opposite to heuristic function, e.g., # non-attacking pairs in 8-queens

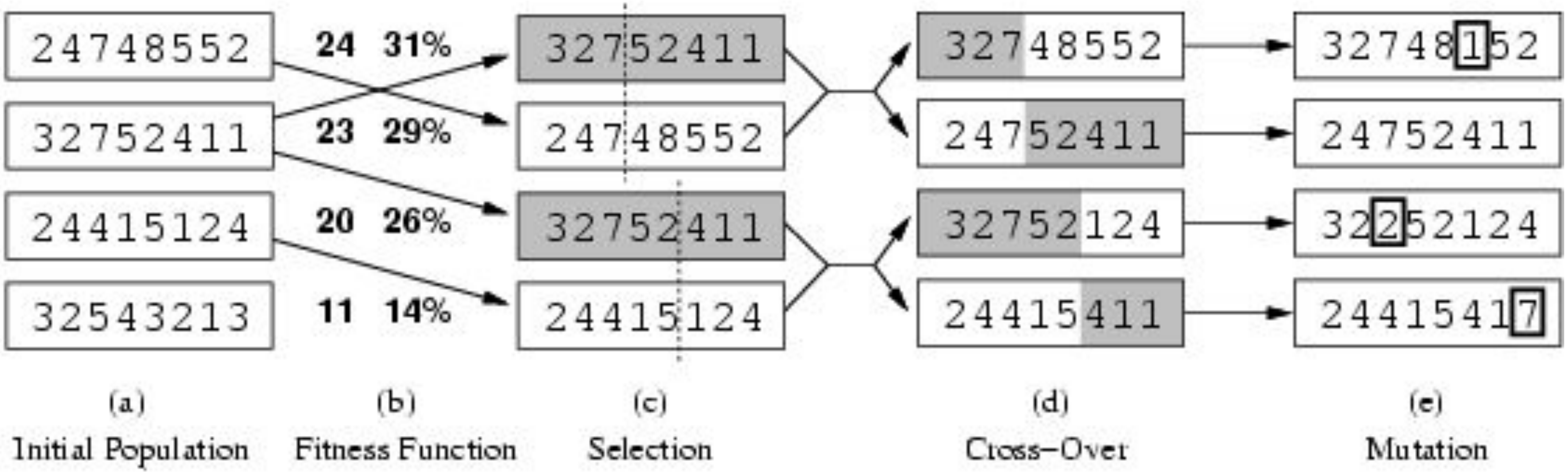
- Produce the next generation of states by “simulated evolution”
  - Random selection
  - Crossover
  - Random mutation



String representation
16257483

Can we evolve 8-queens through genetic algorithms?

Genetic algorithms



4 states for 8-queens problem

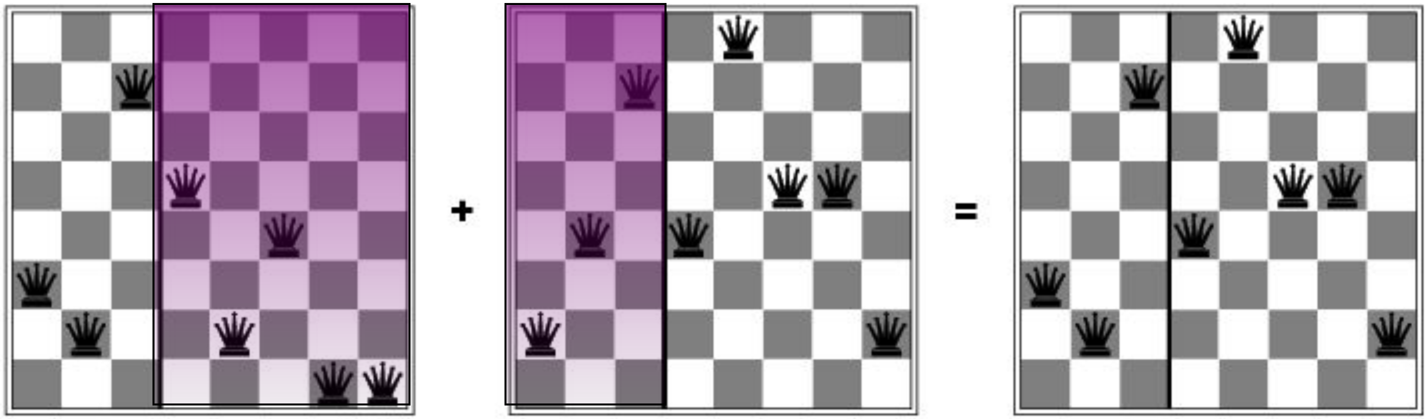
2 pairs of 2 states randomly selected based on fitness. Random crossover points selected

New states after crossover

Random mutation applied

- Fitness function: number of non-attacking pairs of queens (min = 0, max = 8 × 7/2 = 28)
- 24/(24+23+20+11) = 31%
- 23/(24+23+20+11) = 29% etc

Genetic algorithms



Has the effect of "jumping" to a completely different new part of the search space (quite non-local)

Example and code

- Download code in the classroom
- On class: follow a step by step tutorial