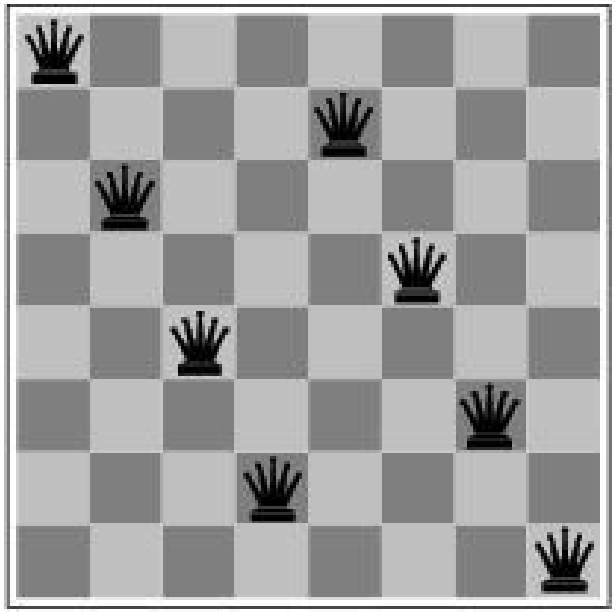
Local Search and Optimization

Outline

- Local search techniques and optimization
  - Hill-climbing
  - Simulated annealing
  - Genetic algorithms
  - Issues with local search

Local search and optimization

- Previous lecture: path to goal is solution to problem
  - systematic exploration of search space.
- This lecture: a state is solution to problem
  - for some problems path is irrelevant.
  - E.g., 8-queens
- Different algorithms can be used
  - Local search



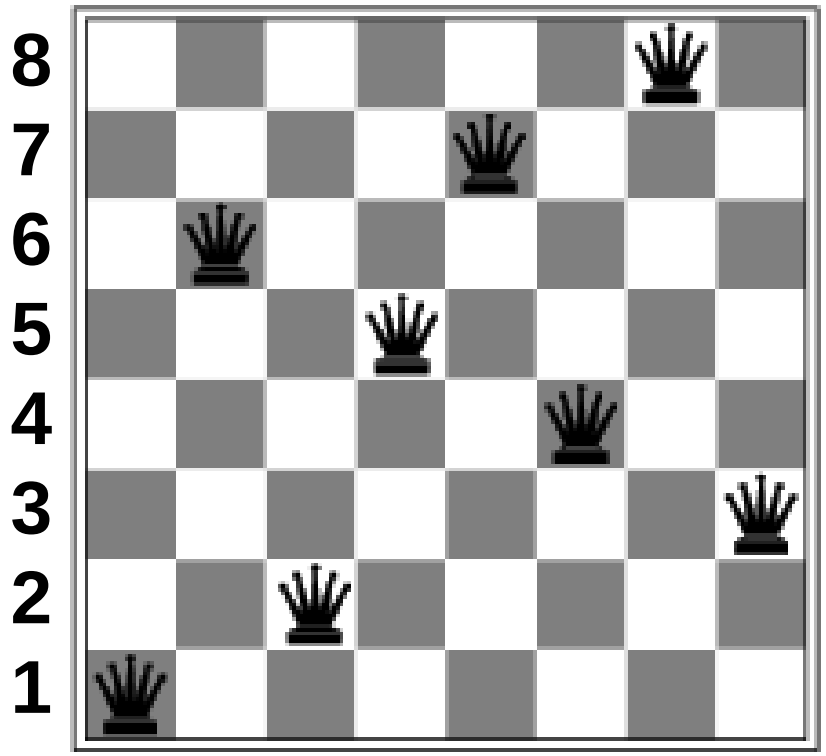
reach the goal node
Constraint satisfaction

optimize(objective fn)
Constraint Optimization

You can go back and forth between the two problems
Typically in the same complexity class

Genetic algorithms

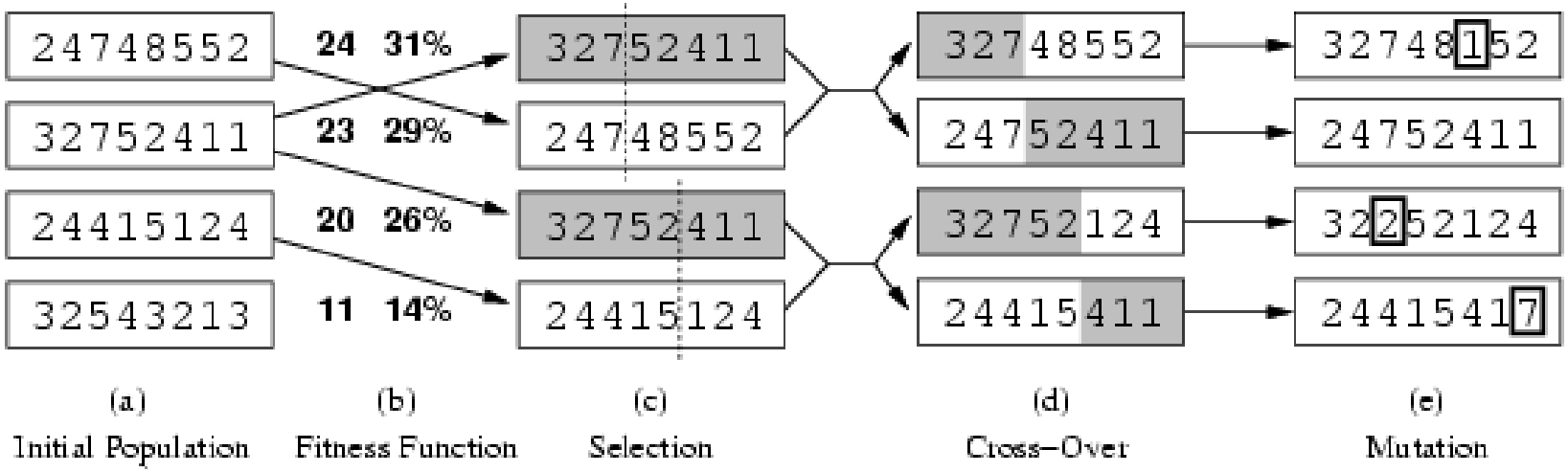
- Twist on Local Search: successor is generated by combining two parent states
- A state is represented as a string over a finite alphabet (e.g. binary)
  - 8-queens
    - State = position of 8 queens each in a column
- Start with *k* randomly generated states (population)
- Evaluation function (fitness 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)