Metaheuristics

2.3 Local Search
2.4 Simulated annealing

Adrian Horga

2.3 Local Search

Local Search

- Other names:
  - Hill climbing
  - Descent
  - Iterative improvement
  - General S-Metaheuristics

- Old and simple method → at each iteration replace the solution with a neighbor if it improves the objective function

Example

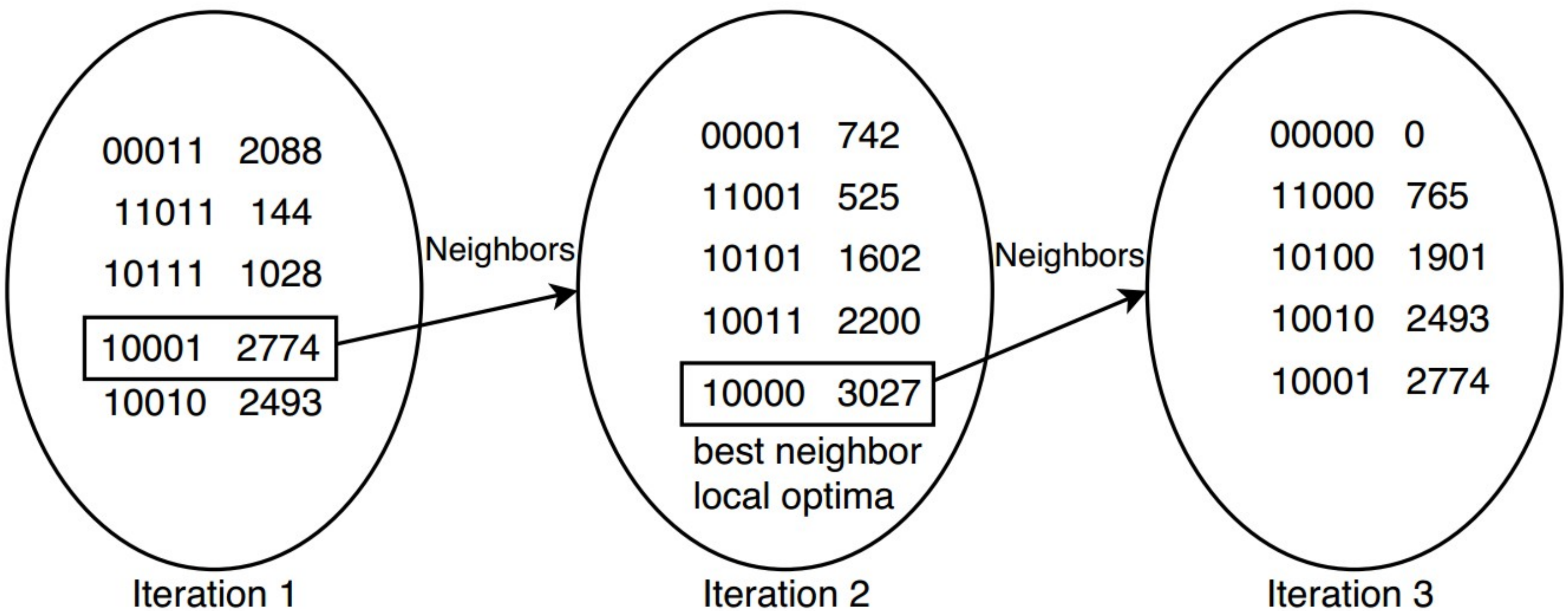

FIGURE 2.21 Local search process using a binary representation of solutions, a flip move operator, and the best neighbor selection strategy. The objective function to maximize is $x^3 - 60x^2 + 900x$. The global optimal solution is $f(01010) = f(10) = 4000$, while the final local optima found is $s = (10000)$, starting from the solution $s_0 = (10001)$.

Another one

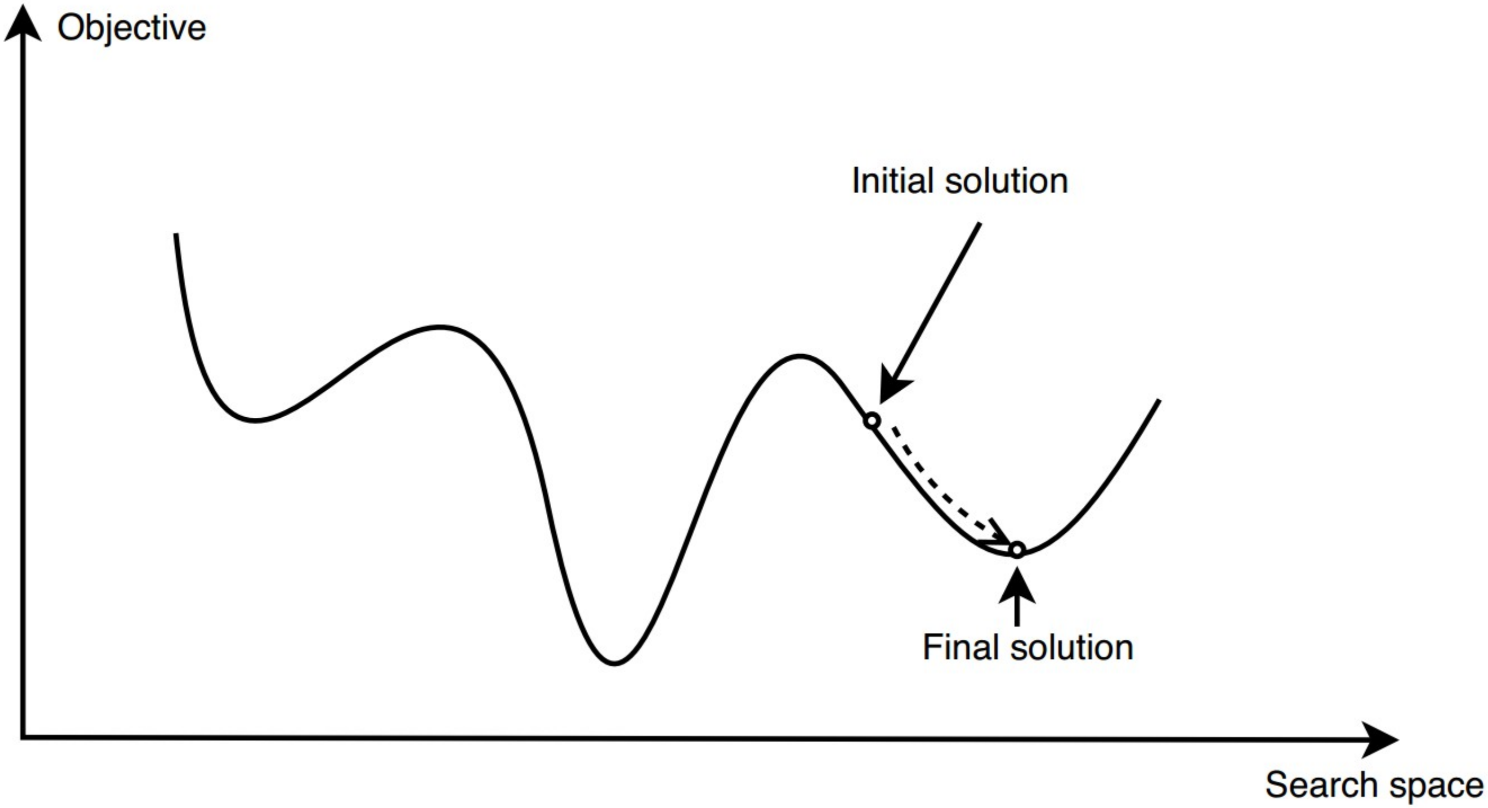

FIGURE 2.22 Local search (steepest descent) behavior in a given landscape.

Properties

- Start with s_0
- Generate k neighbors $(s_1, s_2, \ldots, s_k)$
- “k“ is not known a priori
- $s_{i+1} \in N(s_i), \forall i \in [0, k-1]$
- $f(s_{i+1}) < f(s_i), \forall i \in [0, k-1]$
- $s_k \, is \, a \, local \, optimum: f(s_k) \leqslant f(s), \forall s \in N(s_k)$

Selection of the neighbor – time is money

- Best improvement (steepest descent)
  - Evaluate every neighbor → pick the best
  - Time consuming for large neighborhoods
- First improvement
  - Pick the first that is better
  - In practice is similar to “best improvement”
  - Might need to evaluate everything if no better solution is found
- Random selection
  - Just random

Escaping local optima

- Iterating from different solutions
  - Multistart LS, iterated LS, GRASP
- Accepting non-improving neighbors
  - Simulated annealing
- Changing the neighborhood
  - Variable neighborhood search
- Changing the objective function or the input data of the problem
  - Guided LS, smoothing, noising methods

2.4 Simulated annealing

Simulated annealing

- Based on statistical mechanics → heat then slowly cool a substance to obtain strong structure
- Low starting temperature/fast cooling → imperfections
- SA is a stochastic algorithm which enables degradation of a solution
- Memoryless

Analogy – real life

TABLE 2.4 Analogy Between the Physical System and the Optimization Problem

Physical System	Optimization Problem
System state	Solution
Molecular positions	Decision variables
Energy	Objective function
Ground state	Global optimal solution
Metastable state	Local optimum
Rapid quenching	Local search
Temperature	Control parameter T
Careful annealing	Simulated annealing

Basic idea

- The acceptance probability function → pick nonimproving neighbors
- The cooling schedule → how the temperature decreases (efficiency or effectiveness)
- The higher the temperature → the higher the chance of picking a “bad” neighbor

Move acceptance – or how likely an increase of energy is

$$P(\Delta E, T) = \exp\left(\frac{-\delta E}{kt}\right) > R$$

Cooling schedule

- Initial temperature
- Equilibrium state
- Cooling
- Stopping conditions

Initial temperature - start

- Accept all
  - High starting temperature to accept all neighbors
  - High computation
- Acceptance deviation
  - Use a temperature based on preliminary experimentations
  - Use a standard deviation
- Acceptance ratio
  - Use an interval for the acceptance rate (e.g. [40%, 50%])

Equilibrium state - finish

- Static
  - Predetermined number of transitions to equilibrium state
- Adaptive
  - Characteristics of the search impose the number of generated neighbors
  - Equilibrium state may not be reached at each temperature

Cooling – how do we iterate

- Linear
- Geometric
- Logarithmic
- Very slow decrease
  - Only one iteration per temperature
- Nonmonotonic
  - Temperature may increase again
- Adaptive
  - Dynamic decrease rate
  - Few iter. at high temp. / Many iter. at low temp.

Stopping condition

- Reaching the final temperature
  - Or
- Achieving a predetermined number of iterations
  - Or
- Not improving in a while

Other similar methods

- Threshold accepting
- Record-to-Record Travel
- Great Deluge Algorithm
- Demon Algorithms

Threshold accepting

- Q is the threshold
- Accept only neighbors that are not worse than the threshold
- Ex.: Q may be nonmonotone, or adaptive

Record-to-Record Travel

- “Record” is the best objective values of the visited solutions so far
- “D” accepted deviation
- A small deviation → poor results, faster
- A high deviation → better results, slower

Great Deluge Algorithm

- Analogy with a climber in a rainstorm → rain level goes "UP", the climber needs to keep his feet dry
- Pick neighbor based on water "LEVEL" (above/below)
- Update "LEVEL" based on "UP"

Demon Algorithms – hard to explain

Algorithm 2.8 Template of the demon algorithm.

Input: Demon initial value D
$s = s_0$; /* Generation of the initial solution */
Repeat
  Generate a random neighbor s' ;
  $\Delta E = f(s') - f(s)$;
  If $\Delta E \leq D$ **Then**
    $s = s'$; /* Accept the neighbor solution */
    $D = D - \Delta E$; /* Demon value update */
Until Stopping criteria satisfied
Output: Best solution found.

Demon Algorithms - types

- Bounded DA
- Annealed DA
  - Similar to SA, credit (“D”) is the temperature
- Randomized Bounded DA
  - Use Gaussian distribution for “D”
- Randomized Annealed DA
  - Same search as RBDA with annealing from ADA

Conclusions

- Local search
  - Easy to do, local optima
- Simulated annealing
  - Try to find the best schedule or else you end up doing local search
- Other methods
  - Simulated annealing with less parameters