EE496 : COMPUTATIONAL INTELLIGENCE
EA03 : META-HEURISTICS FOR LOCAL SEARCH

UGUR HALICI
METU: Department of Electrical and Electronics Engineering (EEE)
METU-Hacettepe U: Neuroscience and Neurotechnology (NSNT)

Application of meta heuristics

Meta heuristics can be applied

- on problems where no efficient solving algorithm is known
- e.g. combinatorial optimization problems
- finding an optimal solution is usually not guaranteed
- in comparison with optimal solution: good solutions can be

arbitrarily bad

- success and runtime depends on:
- problem definition and
- implementation of particular steps

⇒ EAs are meta heuristics, too

Local Search Methods

- given: optimization problem $(\Omega, f, \succ)$
- desired: find element $x \in \Omega$, which optimizes f (max. or min.)
- without loss of generality:

  find an element $x \in \Omega$, which maximizes f (should f be mimimized, then we consider $f' \equiv -f$)

⇒ **local search methods** to find local optima in Ω

- **assumption:** f (x1) and f (x2) differ slightly for similar x1, x2 $\in \Omega$

  (no huge jumps in f)
- applicable for arbitrary Ω to find local optima

local search methods

- local search methods = special case of EA
- population: 1 solution candidate ⇒ various consequences
  - recombination operator isn't reasonable as there is only one individual
  - changes: mutation resp. variation operator
  - selection: newly created individual instead of parental individual into next generation?

⇒ one fundamental algorithm for all local search methods

- variants by different acceptance criterion
- individuals contain usually no additional information $Z = \varnothing$
- genotype G depends on problem (as always)

local search

Algorithm 1 fundamental algorithm of local search

Input: objective function F
Output: solution candidate A

```
t ← 0
A(t) ← create solution candidate
evaluate A(t) by F
while termination criterion isn't fulfilled
    B = vary A(t) {
    evaluate B by F
     t ← t + 1
    if Acc(A(t − 1).F, B.F, t) {          /* acceptance criterion, variably implemented */
            A(t) ← B
    } else {
            A(t) ← A(t − 1)
     }
}
return A(t)
```

Gradient Ascent or Descent

Assumption: $\Omega \subseteq IR^n$ and $f : \Omega \rightarrow IR$ is differentiable

- Gradient: differential operation that creates a vector field

⇒ computes vector into the direction of the steepest ascent of the function in a point

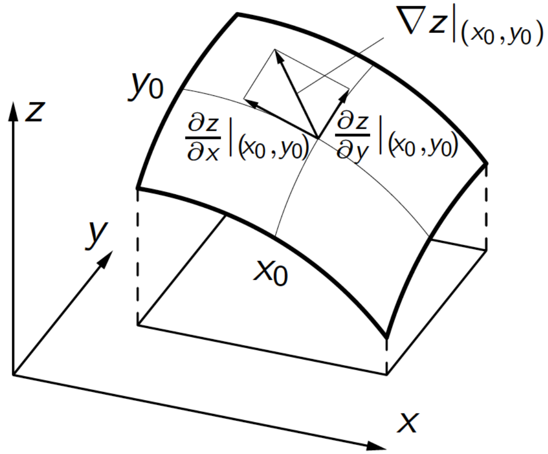


illustration of the gradient of $z = f(x, y)$ at point (x_0, y_0)

$$\nabla z|_{(x_0,y_0)} = \left(\frac{\partial z}{\partial x}|_{(x_0,y_0)}, \frac{\partial z}{\partial y}|_{(x_0,y_0)} \right)$$

Gradient Ascent or Descent

Idea: start at a randomly chosen point, then make small steps in the search space Ω in (or against) the direction of the steepest slope of the function until a (local) optimum is reached

1. Choose a (random) starting point $x(0) = (x^{(0)}_1, \ldots, x^{(0)}_n)$
2. Compute the gradient at the current point $x(^{t)}$

$$\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(t)}\right) = \left(\frac{\partial}{\partial x_1} f\left(\boldsymbol{x}^{(t)}\right), \ldots, \frac{\partial}{\partial x_n} f\left(\boldsymbol{x}^{(t)}\right)\right)$$

3. Make a small step in the direction of the gradient

   $x^{(t+1)} = x^{(t)} + \eta \ \nabla_x f\ (x^{(t)})$

   η: step width parameter ("learning rate" in ANN)

4. Repeat steps 2 and 3 until some termination criterion is fulfilled

   (e.g. user-specified number of steps has been executed, gradient is smaller than a user-specified threshold)

Problems

choice of the step width parameter

- too small value ⇒ large runtime until optimum is reached
- too large value ⇒ oscillations, jump back and forth in Ω
- Solution: momentum term, adaptive step width parameter

Getting stuck in local maxima

- due to local gradient information, maybe only local maxima is reachable
- problem can't be remedied in general
- chance improvement: multiple execution with different starting points

Hill Climbing

Idea: If f is not differentiable, determine direction in which f increases by evaluating random points in the vicinity of the current point

1. Choose a (random) starting point x0 ∈ Ω
2. Choose a point x′ ∈ Ω „in the vicinity" of xt (e.g. by a small random variation of xt)
3. Set

$$x_{t+1} = \begin{cases} x' & \text{if } f(x') > f(x_t), \\ x_t & \text{otherwise} \end{cases}$$

4. Repeat steps 2 and 3 until a termination criterion is fulfilled

Hill Climbing

Pseudocode of acceptance criterion in fundamental algorithm:

Algorithm 2: Acceptance criterion of Hill Climbing

Input: fitness of parent A.F, fitness of offspring B.F, generation t

Output: true or **false**

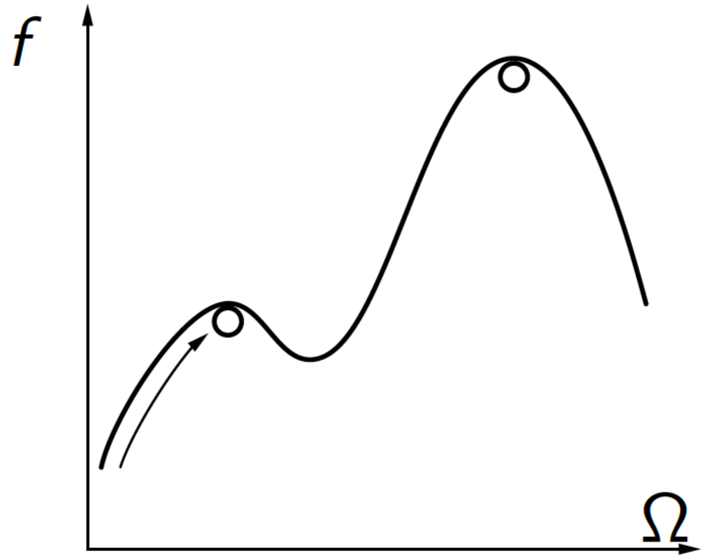
1: **return** B.F > A.F

Problem: Getting stuck in local maxima

- all following methods try to remedy this problem

Simulated Annealing

- Extension of hill climbing and gradient ascent which avoids getting stuck
- **Idea:** passage from lower into higher (local) maxima should be more probable than reversed



Principle:

- random variants of current solution are created
- better solutions are always accepted
- worse solutions are accepted with certain probability which depends on
  - quality difference between current and new solution and
  - temperature parameter (decreases over the time)

Simulated Annealing

Motivation (Minimizing instead of Maximizing)

- physical minimizing of (lattice) energy when heated metal is cooling down slowly
- process is called Annealing
- Purpose: getting softer metal by removing stresses and strains as well as instabilities ⇒ easier metal processing

Alternative motivation: (minimizing as well)

- ball rolls around on irregularly curved surface
- function to minimize: potential energy of ball
- at the beginning: certain kinetic energy overcome slopes
- friction: energy is decreasing ⇒ stopped moving in a valley finally

Attention: no guarantee to find global optimum

Simulated Annealing

1. Choose a (random) starting point x0 ∈ Ω
2. Choose a point x' ∈ Ω „in the vicinity" of current point xt (for example, by a small random variation of xt)
3. Set

$$x_{t+1} = \begin{cases} x' & \text{if } f(x') \geq f(x_t), \\ \left.\begin{array}{l} x' \text{ with probability } p = e^{-\frac{\Delta f}{kT}} \\ x_t \text{ with probability } 1 - p \end{array}\right\} & \text{otherwise} \end{cases}$$

$\Delta f = f(x_t) - f(x')$	quality reduction of the solution
$k = \Delta f_{\max}$	estimate of the range of quality values
T	temperature parameter (decreased over time)

4. Repeat steps 2 and 3 until a termination criterion is fulfilled

   for small T method is almost identical to hill climbing

Simulated Annealing

Algorithm 3 Acceptance criterion of Simulated Annealing

Input: parental fitness $A.F$, fitness of offspring $B.F$, generation t

Output: **true** oder **false**

if $B.F \succ A.F$ {
    return true
} **else** {
    $u \leftarrow$ choose randomly from $U([0,\ 1])$ /* random number between 0 and 1 */
    if $u \leq \exp\left(-\frac{A.F - B.F}{kT_{t-1}}\right)$ {
        return true
    } **else** {
        return false
    }
}

Threshold Accepting

Idea: very similar to simulated annealing, worse solutions are sometimes accepted again, however, with an upper bound for the quality degradation

1. Choose a (random) starting point $x_0 \in \Omega$
2. Choose a point $x' \in \Omega$ „in the vicinity“ of the current point x_t (for example, by a small random variation of x_t)
3. set

$$x_{t+1} = \begin{cases} x' & \text{if } f(x') \geq f(x_t) - \theta, \\ x_t & \text{otherwise.} \end{cases}$$

   θ threshold for accepting worse solution candidates
   (is (slowly) decreased over time)
   (θ = 0 is equivalent to standard hill climbing)

4. Repeat steps 2 and 3 until a termination criterion is fulfilled

---

Algorithm 4 Acceptance criterion of Threshold Accepting

Input: parental fitness A.F, fitness of offspring B.F, generation t

Output: true oder false

1: if B.F > A.F or A.F − B.F ≤ θ {

2:    **return true**

3: } **else** {

4:    **return false**

5: }

Great Deluge Algorithm (Büyük Tufan : Nuh) ☺

Idea: very similar to simulated annealing, worse solutions are sometimes accepted again, absolute lower bound is used

1. Choose a (random) starting point x0 ∈ Ω
2. Choose a point x′ ∈ Ω "in the vicinity" of the current point xt (e.g. by a small random variation of xt)
3. Set

$$x_{t+1} = \begin{cases} x' & \text{if } f(x') \geq \theta_0 + t \cdot \eta, \\ x_t & \text{otherwise} \end{cases}$$

θ_0 lower bound for the quality of the candidate solutions at t = 0 (initial "water level")

η step width parameter ("speed of the rain")

4. Repeat steps 2 and 3 until a termination criterion is fulfilled

---

Algorithm 5 Acceptance criterion of Great Deluge Algorithm

Input: parental fitness A.F, fitness of offspring B.F, generation t

Output: true oder false

1: **if** B.F $> \theta_0 + \eta \cdot t$ {

2:     **return true**

3: } e**lse** {

4: **return false**

5: }

Record-to-Record Travel

Idea: similar to great deluge algorithm, rising water level is used, linked to the fitness of the best found individual

- possbile degradation: always seen in relation to the best found individual
- only if there is an improvement: current individual is important
- similar to threshold accepting: a monotonously increasing sequence of real numbers controls the selection of poor individuals

Record-to-Record Travel

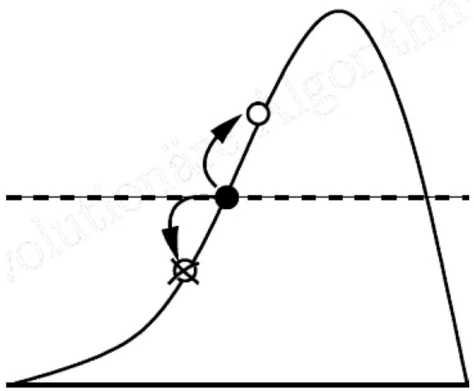
Algorithm 6 Acceptance criterion of Record-to-Record Travel

Input: parental fitness A.F, fitness of offspring B.F, t, best found quality Fbest

Output: **true** or **false**, Fbest

1: **if** B.F > Fbest {
2:     Fbest ← B.F
3:     **return true**, Fbest
4: } **else** {
5:     **if** B.F > F_{best} - T_t {
6:         **return true**, Fbest
7:     }
8: }
9: **return false**, Fbest

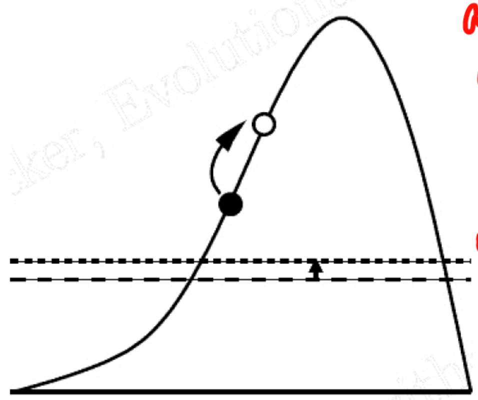
Comparison of local search algorithms



Hill Climbing

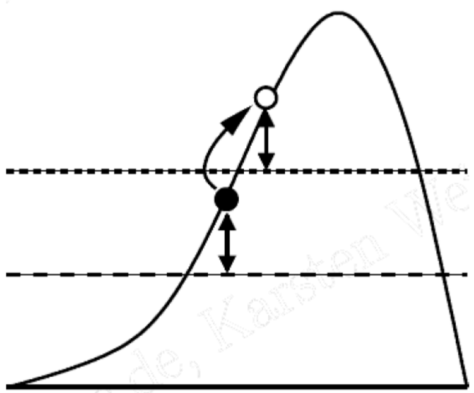
accepted only if new is better than prev

in simulated annealing both are accepted probabilistically



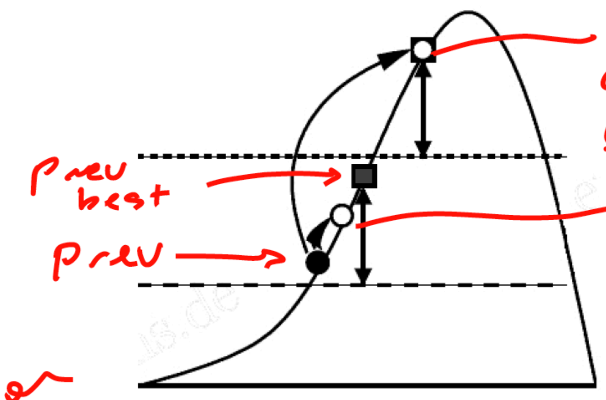
Great Deluge Algorithm

accepted only if new is better than increasing threshold



Threshold Accepting

accepted only if new is better than prev − θ, and then new becomes prev in the next iteration



Record-to-Record Travel

new accepted and set ~~as~~ best

new accepted but best remains the same