Optimization

Homayoun Valafar

Department of Computer Science and Engineering, USC

Optimization and Protein Folding

- Theoretically, the structure with the minimum total energy is the structure of interest
- Total energy is defined by the force-field

$$E_{Total} = \sum \left[w^p_{BOND} E_{BOND} + w^p_{ANGL} E_{ANGL} + w^p_{DIHE} E_{DIHE} + w^p_{IMPR} E_{IMPR} + w^p_{VDW} E_{VDW} + w^p_{ELEC} E_{ELEC} \right]$$

- The core of Ab Initio protein folding is optimization
- A robust optimization method is cruicial for successful protein folding algorithms

Optimization Problem

- Given an objective (cost) function f(x) find the optimal point X* such that:

$$\begin{cases} X, X^* \in R_n \\ f(X): R_n \rightarrow R \\ f(X^*) \leq f(X) \; \forall X \end{cases}$$

- Optimization is the root of most computational problems
- Many different approaches with their unique set of advantages and disadvantages

Monte Carlo

- Easiest to implement
- Very effective for low dimensional problems
- Very ineffective for large dimensional problems
- Algorithm consists of randomly sampling space and accepting the point X* with the smallest value of objective function

```
for i=1 to 10000 {
    X = random(range)
    If f(X) <= f(X*) then X*=X
}
```

Gradient Descent
(Conjugate Gradient, Hill Climbing)

- Start with some initial point X_0
- Calculate X_{k+1} from X_k in the following way:

$$x_{k+1} = x_k \pm \rho . \nabla f(x_k)$$

- Here ρ is the descent step size parameter
  - ρ needs to be selected carefully. Too small or too large can have severe consequences.
- Calculate $f(X_{k+1})$
- Go to step 2.

Gradient of A Function

- A multi-dimensional derivative
- For an n-dimensional function produces an n-dimensional vector pointing at the direction of highest increase
- Following the direction of the gradient will increase f(x) optimally
- Following in the opposite direction of the gradient will decrease f(x) optimally
- Example:

$$f(x,y,z)=x^2y+xyz+y^3+3xy\sqrt{z}$$

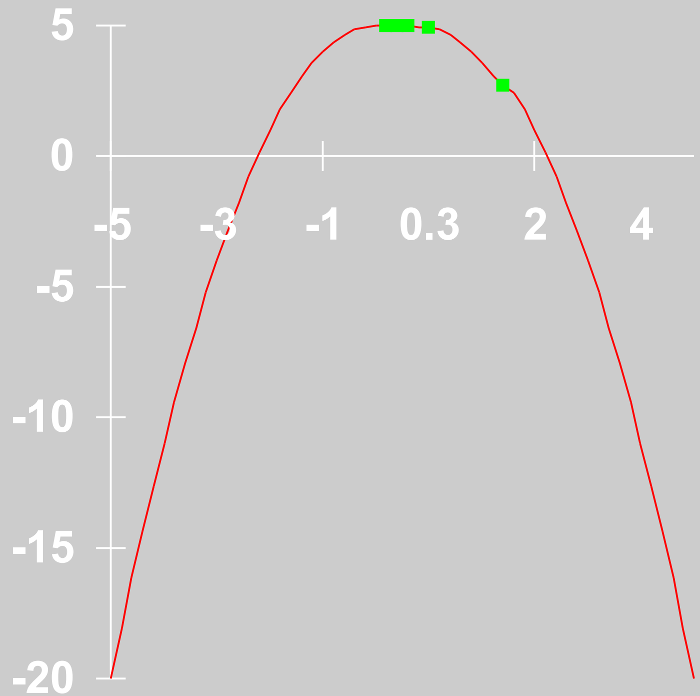
$$\nabla\vec{f}(x,y,z)=\begin{pmatrix} 2xy+yz+3y\sqrt{z} \\ x^2+xz+3y^2+3x\sqrt{z} \\ xy-\dfrac{3xy}{2\sqrt{z}} \end{pmatrix}$$

Example of Gradient ascend

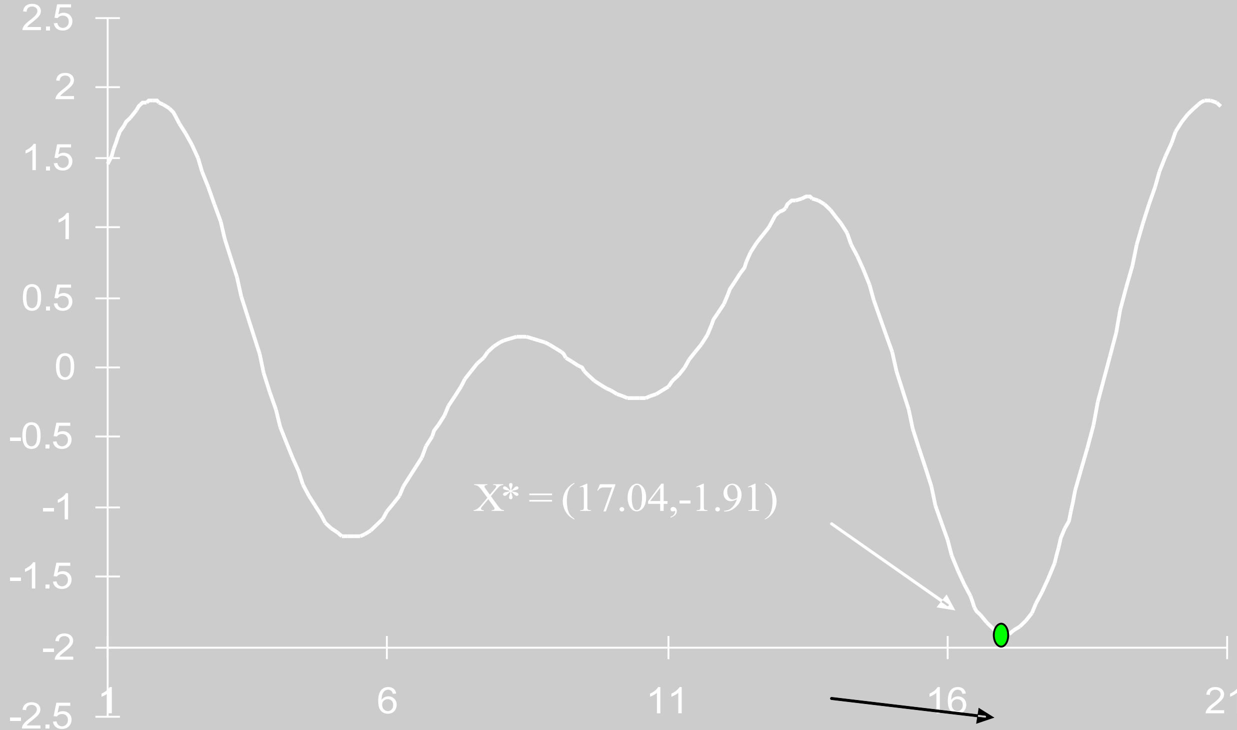
$f(x) = -x^2 - 5;$

$x_{max} = 0, f_{max} = 5$

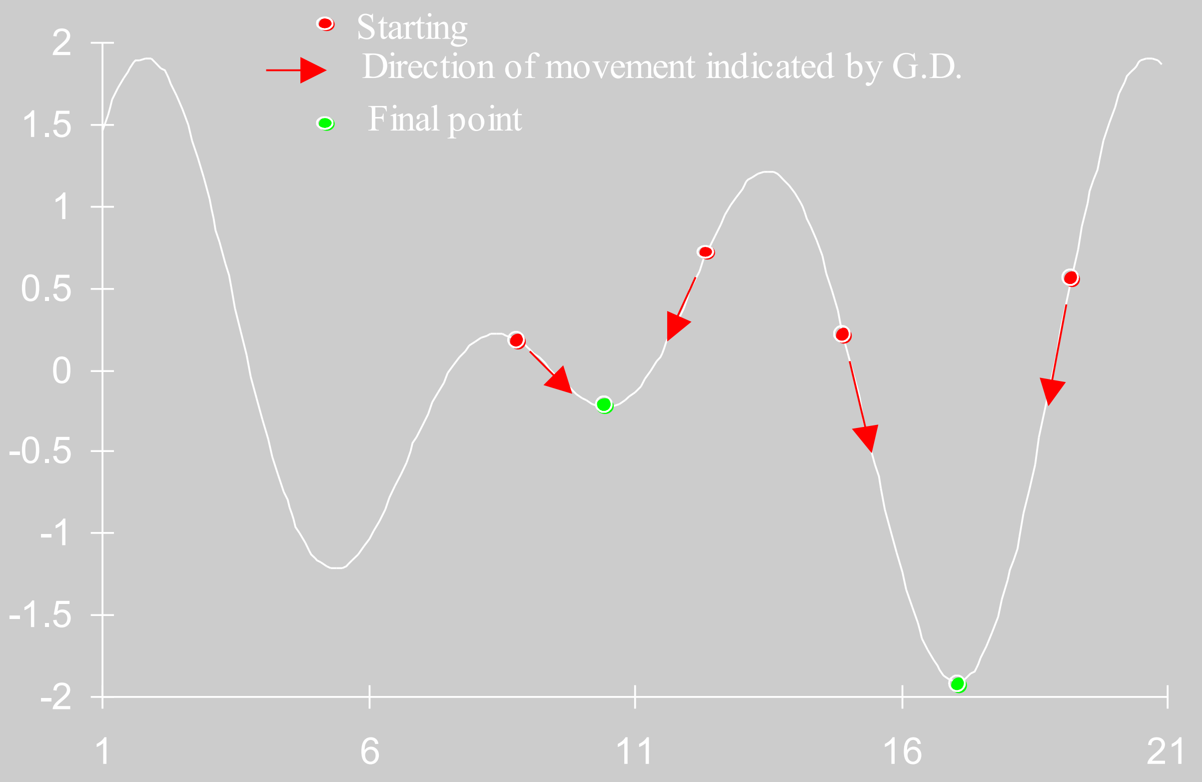
x_k	$\nabla f'(x_k) = -2x$	$x_{k+1} = x_k + 0.4 * \nabla f'(x_k)$
1.5	-3	0.3
0.3	-0.6	0.06
0.06	-0.12	0.012
0.012	-0.024	0.0024



Example of an Objective (Cost) Function



Gradient Descent



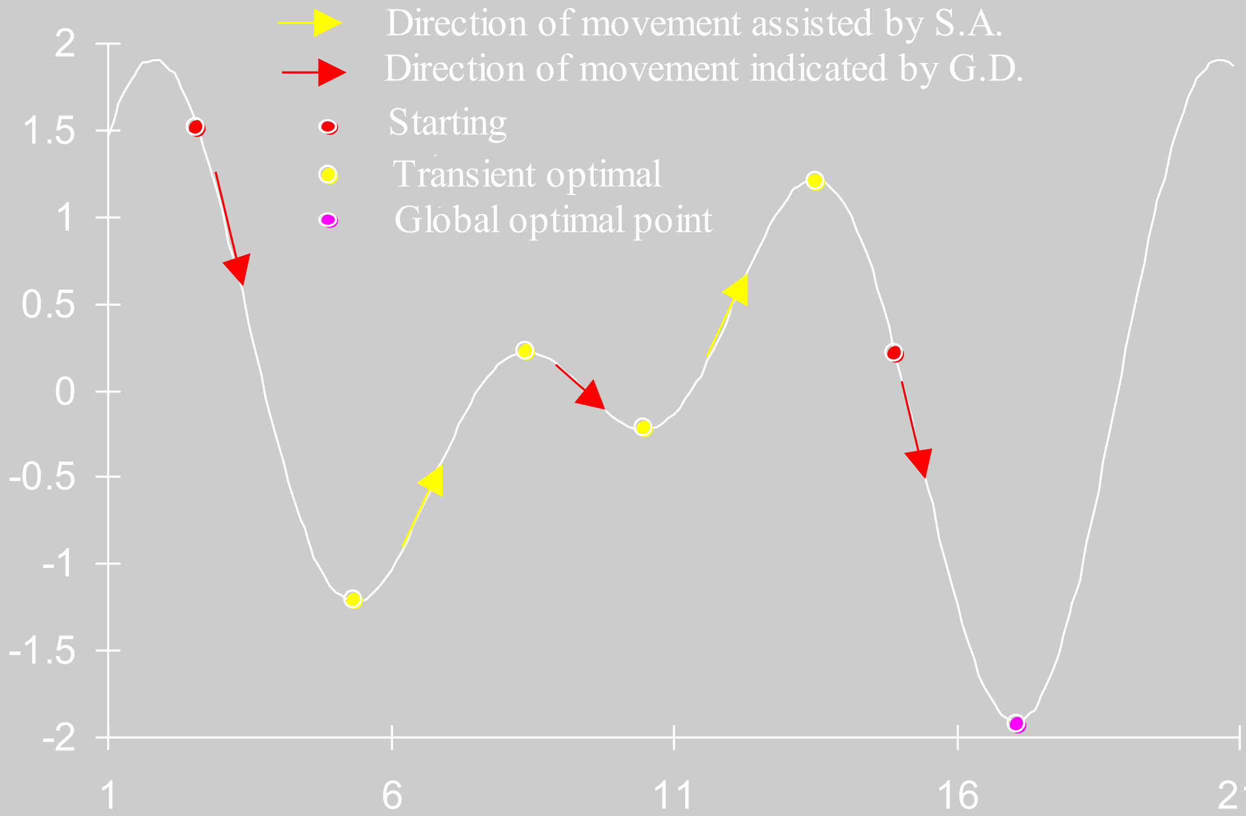
Simulated Annealing Metropolis Algorithm

Pseudo PASCAL code

```
Initialize(i_stat , T_0, L_0);
k := 0; i := i_stat ;
repeat
    for i := 1 to L_k do
    begin
        Generate(X_j from X_i);
        if f(j) < f(i) then i := j;
        else
            if (exp(f(i) -f(j))/T_k) > random[0,1) then i := j
        end;
        k := k+1;
        Calculate_Control(T_K);
    end;
    Calculate_Length(L_k);
until stop criterion
```

Contribution of Simulated Annealing

Simulated annealing helps to escape from the local minima.



Limited Success with GD