PROBLEMI DI OTTIMIZZAZIONE

In AI, uno degli strumenti principali sono gli algoritmi di ricerca dell'ottimo.

Memetic Computing

Sono algoritmi di ottimizzazione affiancati da tecniche probabilistiche che non richiedono le assunzioni di linearità e convessità del problema.

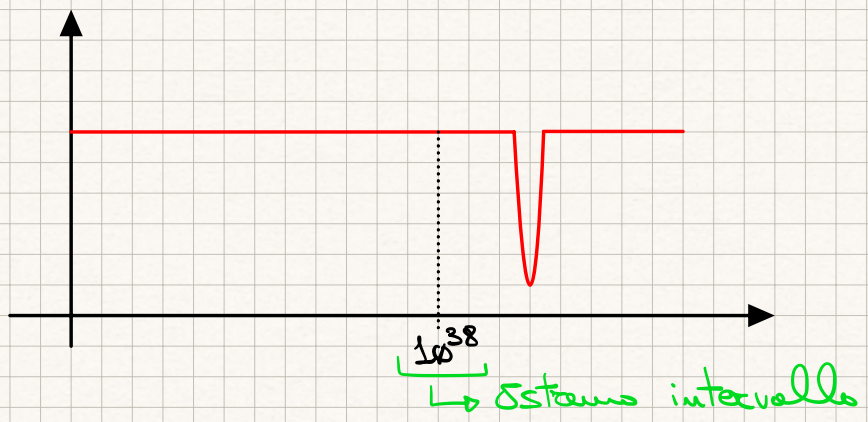
Trovare l'ottimo globale è davvero difficile. È molto più semplice trovare l'ottimo locale.

Bisogna dividere due modi di ricerca dell'ottimo globale:

1) Unconstrained Optimization
2) Box-Constrained Optimization

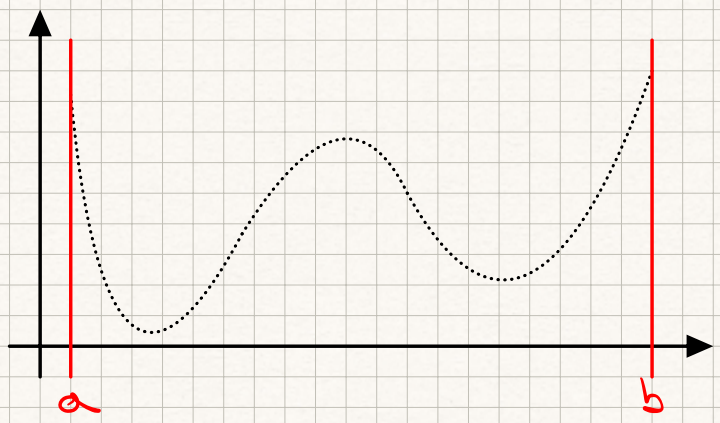
Esempio Unconstrained

Se sulla nostra macchina usiamo lo standard IEEE 754 siamo limitati nella rappresentazione di un numero, abbiamo difatti un intervallo. E se l'ottimo cade fuori da questo intervallo? Non lo troviamo



Esempio Box-Constrained

Molto più semplice trovare l'ottimo, ma spesso è un ottimo locale:



Random Search

Algoritmo di ottimizzazione che salta randomicamente da un punto a un altro in cerca dell'ottimo.
Funziona male se abbiamo tanti dati da gestire.

SIMULATED ANNEALING (SA)

Algoritmo di ottimizzazione che cerca di approssimare l'ottimo globale di una funzione, in un ampio spazio di ricerca.
Spesso usato quando lo spazio di ricerca è discreto.

Pseudo-Codice

```
S = S0
for (k = 0 ... kmax-1)
{
    T = temperature(1 - (k+1)/kmax);
    vicino = random-neighbour();
    Snew = state-of(vicino)
    if (P(δ(S), δ(Snew), T) ≥ random(0, 1))
    {
        S = Snew
    }
}
return S;
```

L'algoritmo testa il valore della funzione nel punto attuale con il valore in un punto vicino. Se il nuovo valore è migliore del primo, nel senso che si riesce a fare un passo verso l'ottimo, allora con una certa probabilità viene aggiornato il punto ottimo col nuovo punto.

PROBABILITÀ

Questa probabilità di aggiornamento è

$$P(\delta(s), \delta(s_{new}), T)$$

Dipendente da $\delta(s)$, $\delta(s_{new})$ e T.
Il parametro di temperatura T, si raffredda man mano l'algoritmo prosegue (diminuisce). Questo fa si che, più T scende più l'algoritmo si comporta come un algoritmo greedy che accetta ogni passo migliore di quello attuale.
Quando T è alto, si accettano con una probabilità non nulla, anche passi sfavorevoli così da evitare di rimanere incastrati in un ottimo locale.

Pseudo-Codice

```
function [best_x, best_y] = SA_RE_1D(f,a,b,niter,lambda,dispAndPlot)
    time = 1:niter;
    temperature = exp(-lambda*time);

    x1 = rand()*(b-a)+a; % unifrnd(1,1,a,b)
    y1 = feval(f,x1); % f(x)

    current_x = x1; % current solution, to be perturbed
    current_y = y1; % value of the objective function of the current solution

    best_x = x1;
    best_y = y1;

    for i=2:niter
        new_x = perturb(current_x,a,b);
        new_y = feval(f,new_x);

        % if the temperature is greater than a random number in [0,1]
        if (new_y < current_y) || ( rand() < temperature(i) )
            current_x = new_x;
            current_y = new_y;
        end
        if new_y < best_y
            best_x = new_x;
            best_y = new_y;
        end
    end
end
```

```
function new_x = perturb(current_x,a,b)
    % perturb the current solution, by adding a random number in the interval [-0.1,0.1]
    new_x = current_x + 0.2*(b-a)*(rand()-0.5);

    % check if the new solution is within the bounds
    if new_x > b
        new_x = b;
    end
    if new_x < a
        new_x = a;
    end
end
```

Esempio perturbazione

$[a, b] = [3, 6]$ $\quad x_{curr} = 5$

$$x_{new} = 5 + \frac{0.2 \cdot (3)(0.8 - 0.5)}{0.18} = 5.18$$