Playground

Abdelrahman

September 22, 2024

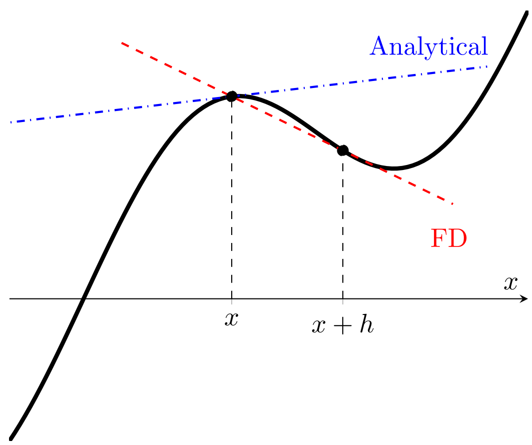


Figure 1: Forward Finite Difference

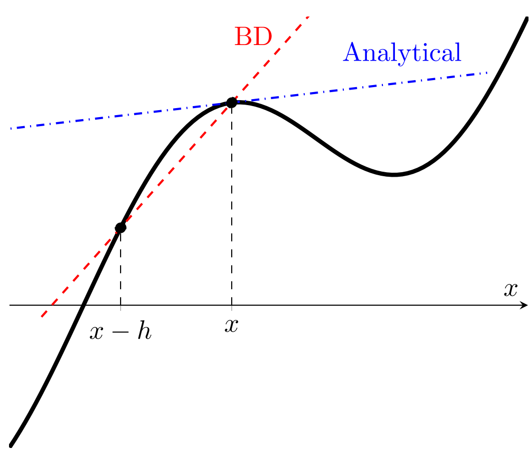


Figure 2: Backward Finite Difference

Algorithm 1: Hill Climbing Algorithm

Data: x_i
Result: $x_n \leftarrow$ last solution candidate

$x_0 \leftarrow$ starting solution candidate;

for $i \in \{0, \ldots, n\}$ **do**
    $x_{\text{neighbor}} \leftarrow$ select best neighbor from x_i neighborhood;
    $\Delta\, cost \leftarrow$ compute cost difference between x_{neighbor} and x_i;

    if cost(x_{neighbor}) is better than cost(x_i) **then**
        Accept $x_{i+1} \leftarrow x_i$;

    else if cost(x_{neighbor}) is worse than cost(x_i) **then**
        Terminate Search;
end

Algorithm 2: Simulated Annealing Algorithm

Data: T_i, x_i
Result: $x_n \leftarrow$ last solution candidate

$T_0 \leftarrow$ initial temperature of cooling schedule;
$x_0 \leftarrow$ starting solution candidate;

for $i \in \{0, \ldots, n\}$ **do**
    $x_{\text{neighbor}} \leftarrow$ randomly sample a neighbor from x_i neighborhood using uniform distribution;
    $\Delta\, cost \leftarrow$ compute cost difference between x_{neighbor} and x_i;

    if cost(x_{neighbor}) is better than cost(x_i) **then**
        Accept $x_{i+1} \leftarrow x_i$;

    else if cost(x_{neighbor}) is worse than cost(x_i) **then**
        if probability $e^{-\Delta cost\, /\, T_i}$ **then**
            Accept $x_{i+1} \leftarrow x_i$;
        else
            Reject Do nothing;
        end
end

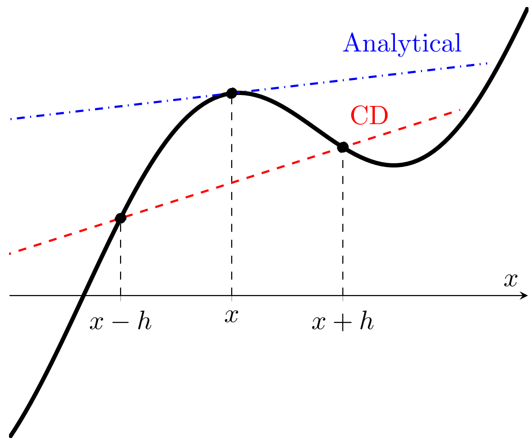


Figure 3: Central Finite Difference

Assumptions	
Steady & Fully developed $\Rightarrow$ 1	$\frac{\partial}{\partial t} = 0\ , \quad \frac{\partial}{\partial x} = 0$
No-Slip Condition $\Rightarrow$ 2	$\mathbf{u}(x,0) = \mathbf{0}\ , \quad \mathbf{u}(x,d) = \begin{bmatrix} U \\ 0 \end{bmatrix}$
Constant Pressure $\Rightarrow$ 3	$\frac{\partial p}{\partial x} = 0\ , \quad \frac{\partial p}{\partial y} = 0$
Newtonian Fluid	constant viscosity (ν)
Incompressible	constant density (ρ)
Laminar & Purely axial	$v = 0$

Assumptions	
Steady & Fully developed $\Rightarrow$ 1	$\frac{\partial}{\partial t} = 0\ , \quad \frac{\partial}{\partial x} = 0$
No-Slip Condition $\Rightarrow$ 2	$\mathbf{u}(x,0) = \mathbf{0}\ , \quad \mathbf{u}(x,d) = \mathbf{0}$
Constant Pressure $\Rightarrow$ 3	$\frac{\partial p}{\partial x} = -G\ , \quad \frac{\partial p}{\partial y} = 0$
Newtonian Fluid	constant viscosity (ν)
Incompressible	constant density (ρ)
Laminar & Purely axial	$v = 0$

Assumptions	
Steady & Fully developed $\Rightarrow$ 1	$\frac{\partial}{\partial t} = 0\ , \quad \frac{\partial}{\partial x} = 0$
No-Slip Condition $\Rightarrow$ 2	$\mathbf{u}(x,y,z) = \mathbf{0} \quad \Longleftrightarrow \quad \sqrt{y^2+z^2} = R$
Constant Pressure $\Rightarrow$ 3	$\frac{\partial p}{\partial x} = -G\ , \quad \frac{\partial p}{\partial y} = 0\ , \quad \frac{\partial p}{\partial z} = 0$
Newtonian Fluid	constant viscosity (ν)
Incompressible	constant density (ρ)
Laminar & Purely axial	$v = 0\ , \quad w = 0$

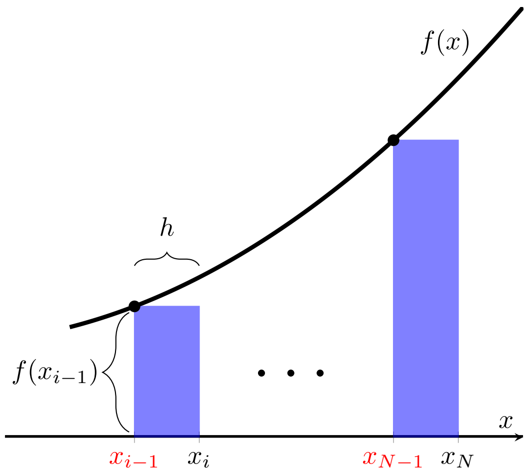


Figure 4: Rectangle Method Left Point Rule

References

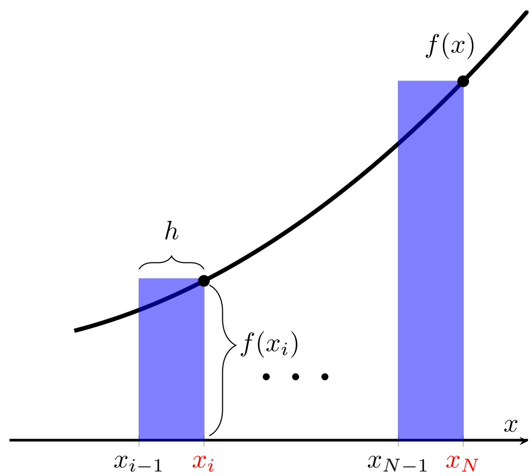


Figure 5: Rectangle Method Right Point Rule

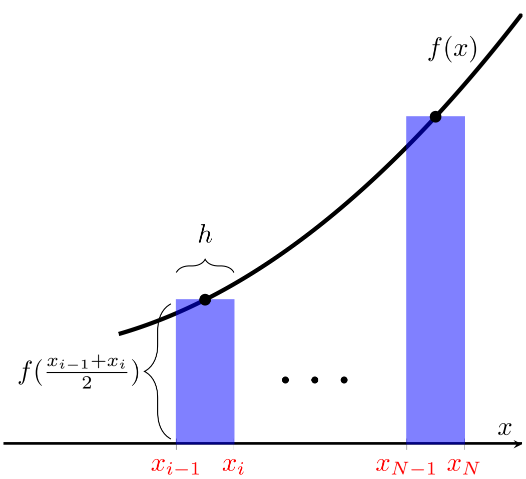
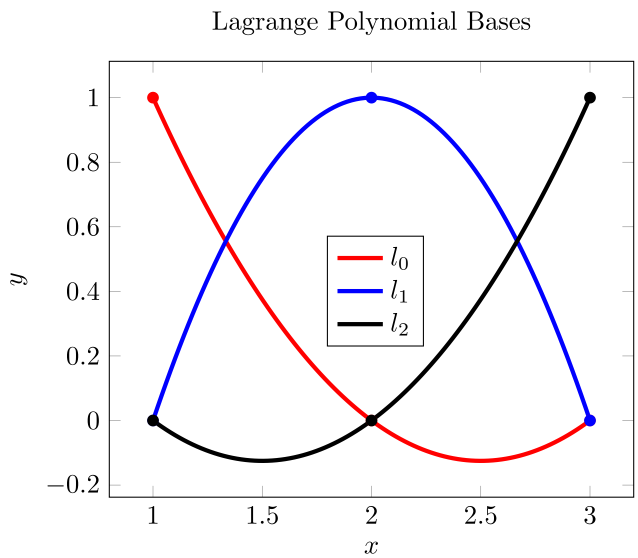


Figure 6: Rectangle Method Midpoint Rule



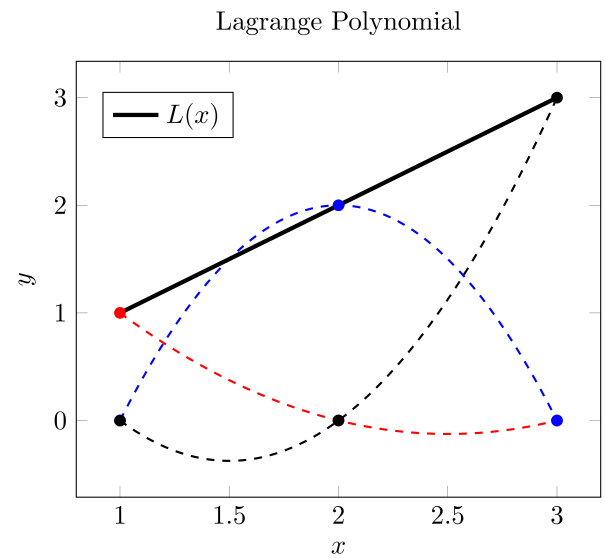


Figure 7: Lagrange Polynomial and its Bases

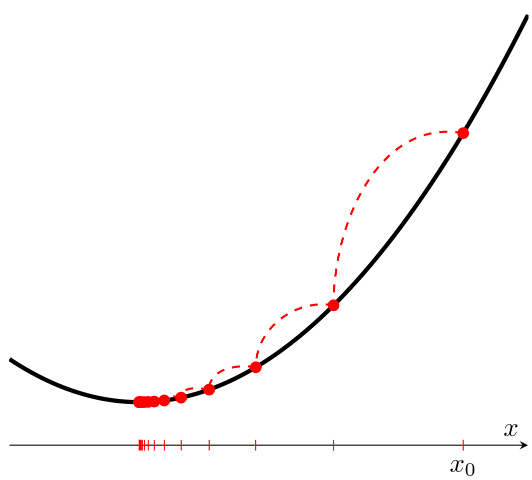


Figure 8: Gradient Decent Steps

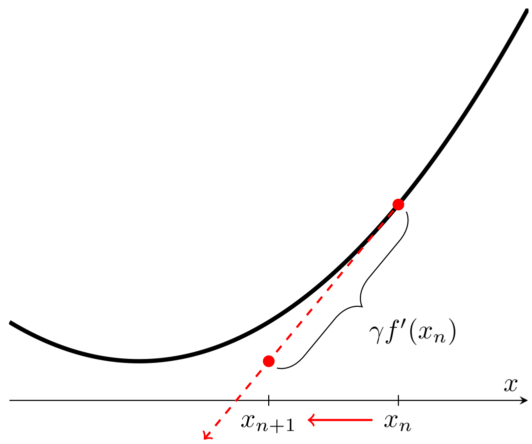


Figure 9: Gradient Decent Step

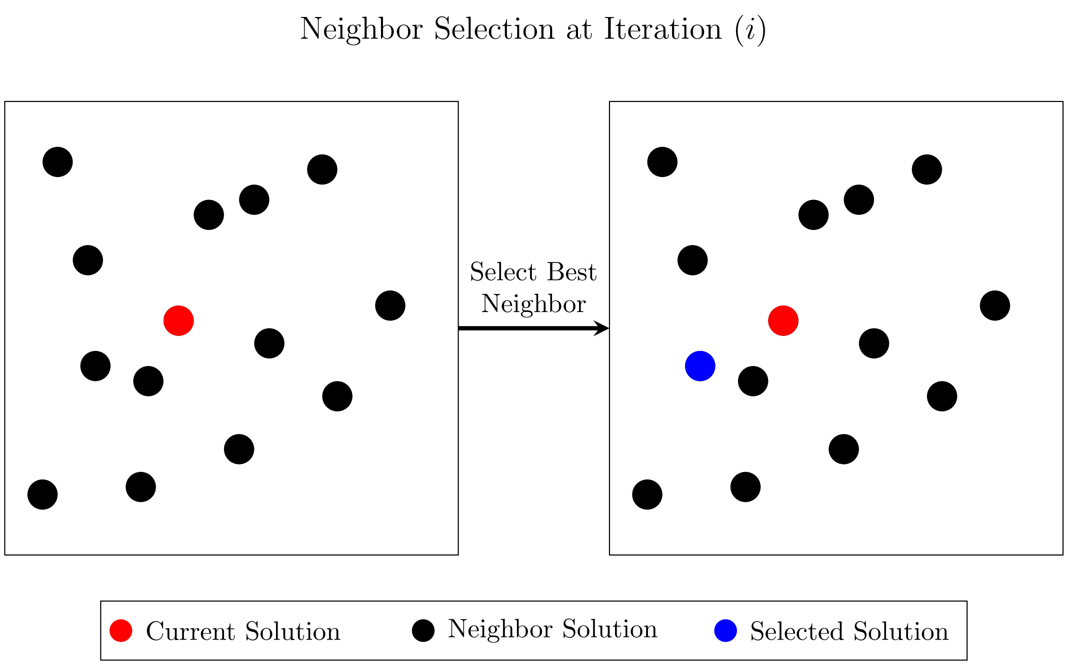


Figure 10: Hill Climbing Neighbor Selection

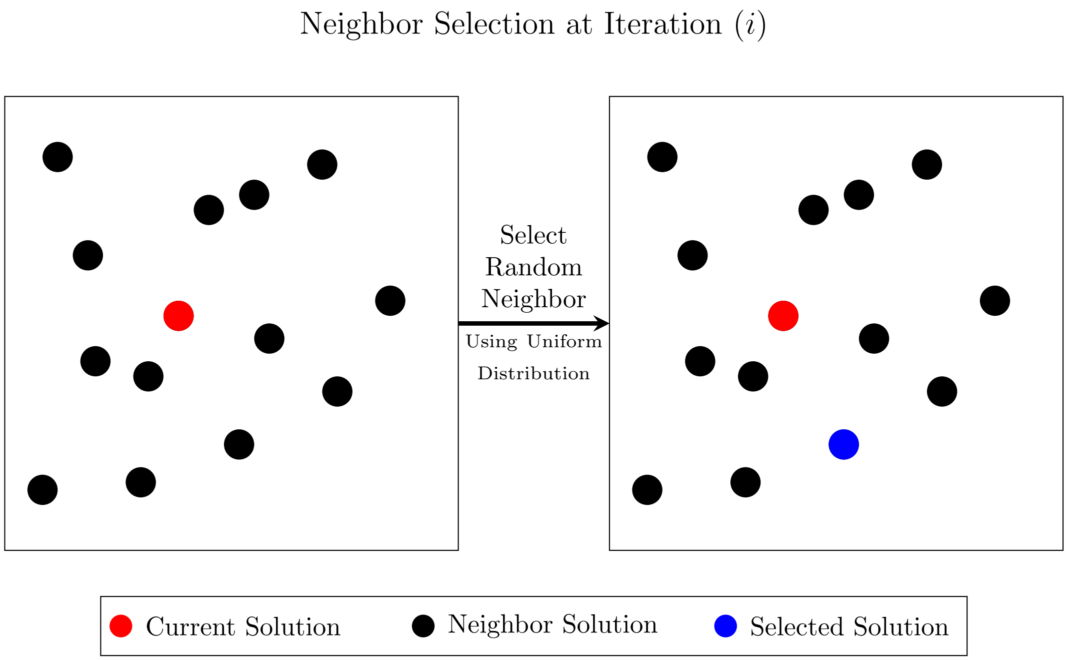


Figure 11: Simulated Annealing Neighbor Selection

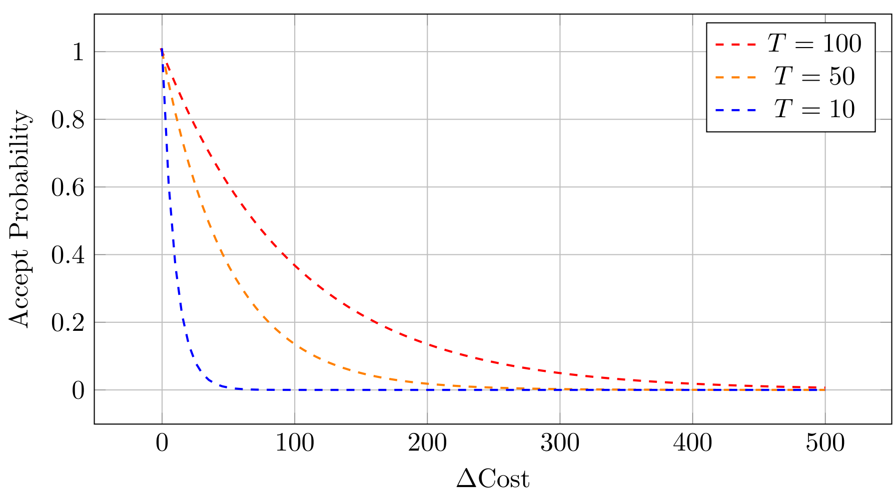


Figure 12: Simulated Annealing Temperature

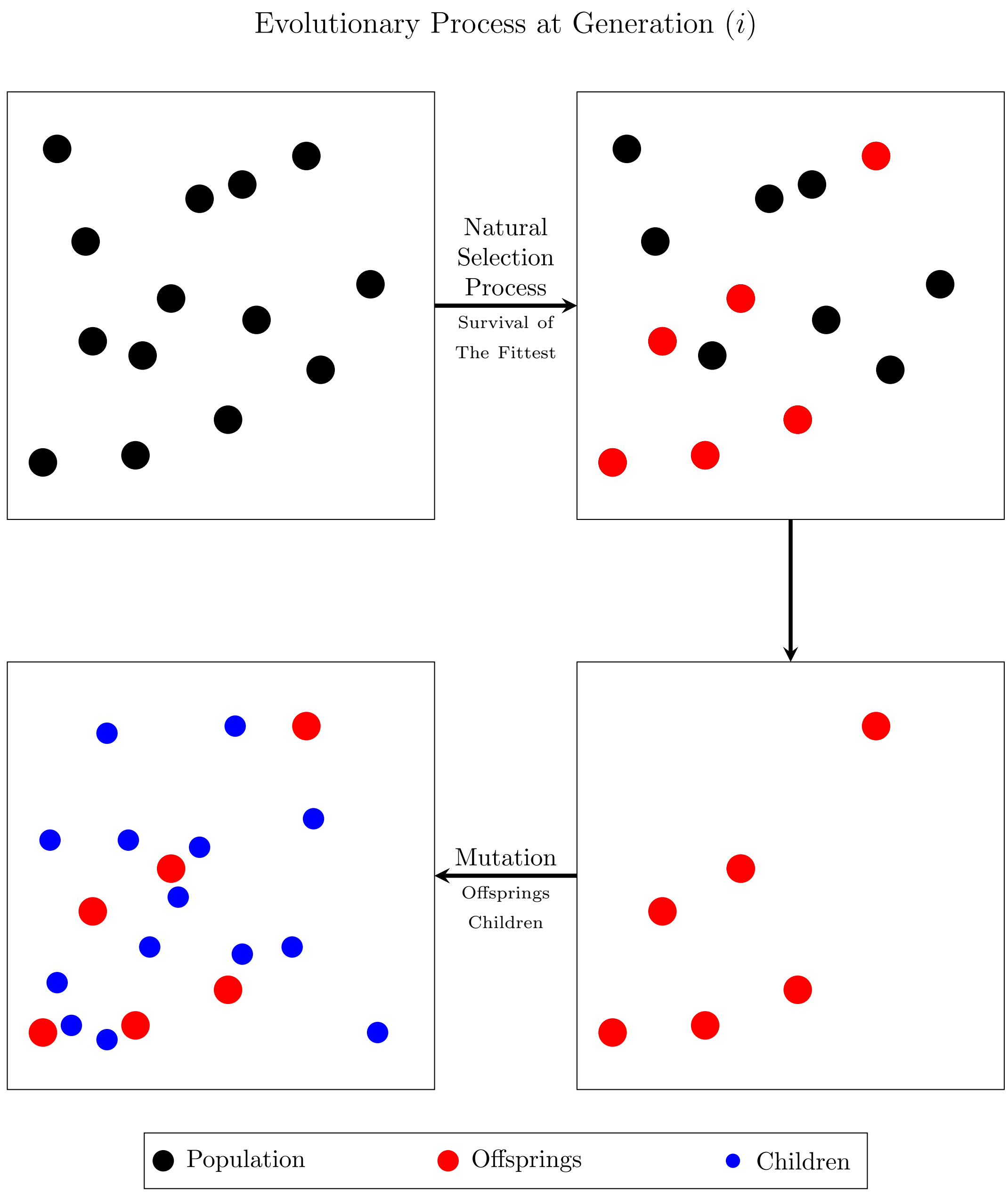


Figure 13: Evolutionary Process in Evolutionary Strategies Algorithm

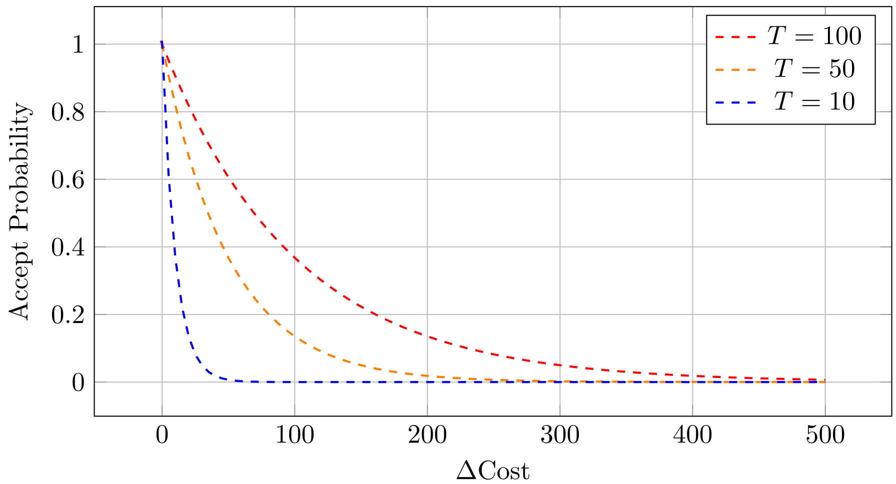


Figure 14: Simulated Annealing Temperature

Algorithm 3: Simulated Annealing Algorithm

Data: T_i, x_i
Result: $x_n \leftarrow$ last solution candidate

$T_0 \leftarrow$ initial temperature of cooling schedule;
$x_0 \leftarrow$ starting solution candidate;

for $i \in \{0, \ldots, n\}$ **do**
  $x_{\text{neighbor}} \leftarrow$ randomly sample a neighbor from x_i neighborhood using uniform distribution;
  $\Delta\, cost \leftarrow$ compute cost difference between x_{neighbor} and x_i;
  if cost(x_{neighbor}) is better than cost(x_i) **then**
    Accept $x_{i+1} \leftarrow x_i$;
  else if cost(x_{neighbor}) is worse than cost(x_i) **then**
    if probability $e^{-\Delta cost\, /\, T_i}$ **then**
      Accept $x_{i+1} \leftarrow x_i$;
    else
      Reject Do nothing;
    end
end

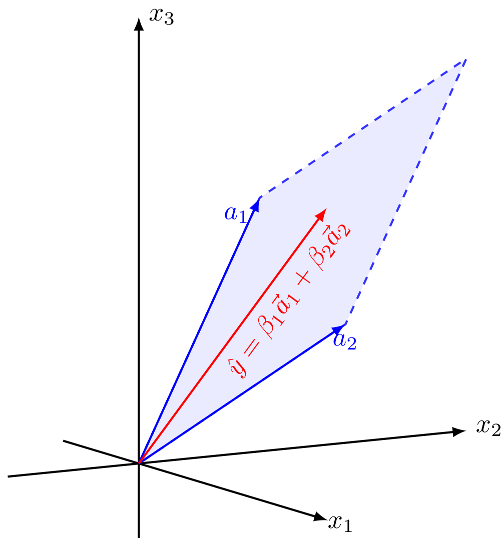


Figure 15: Least Square Problem - Linear combination of column vectors

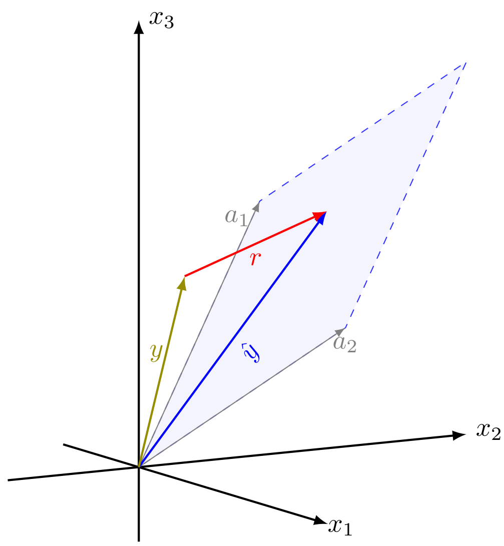


Figure 16: Least Square Problem

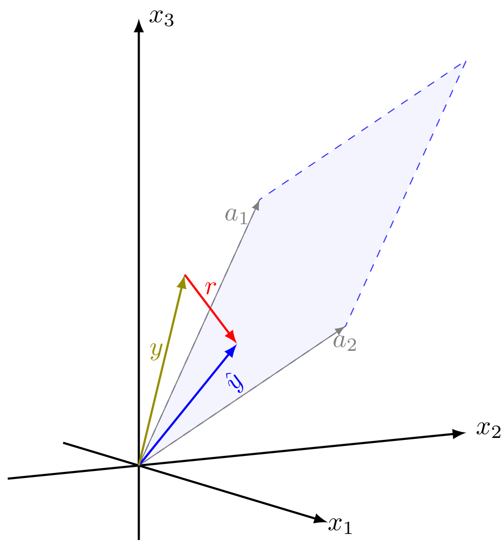


Figure 17: Least Square Problem