LOCAL SEARCH METAHEURISTICS

Presented by:

- **Chakir Hicham**
- **Echamkh Yassine**
- **Akaabour Oussama**

Supervised by:

- **Pr. Aziz OUAARAB**

CONTENT

CONTEXT

Why Local search ?

- The local search method is an approach used in computer science, mathematics, and other fields to find an optimal solution to a problem by **iteratively** exploring the **solution space** starting from an **initial solution**

- Here are some examples of fields where this method is used :

  ➡ Mathematical optimization

  ➡ Routing problems

  ➡ Design and engineering problems

  ➡ Games and solving strategic problems

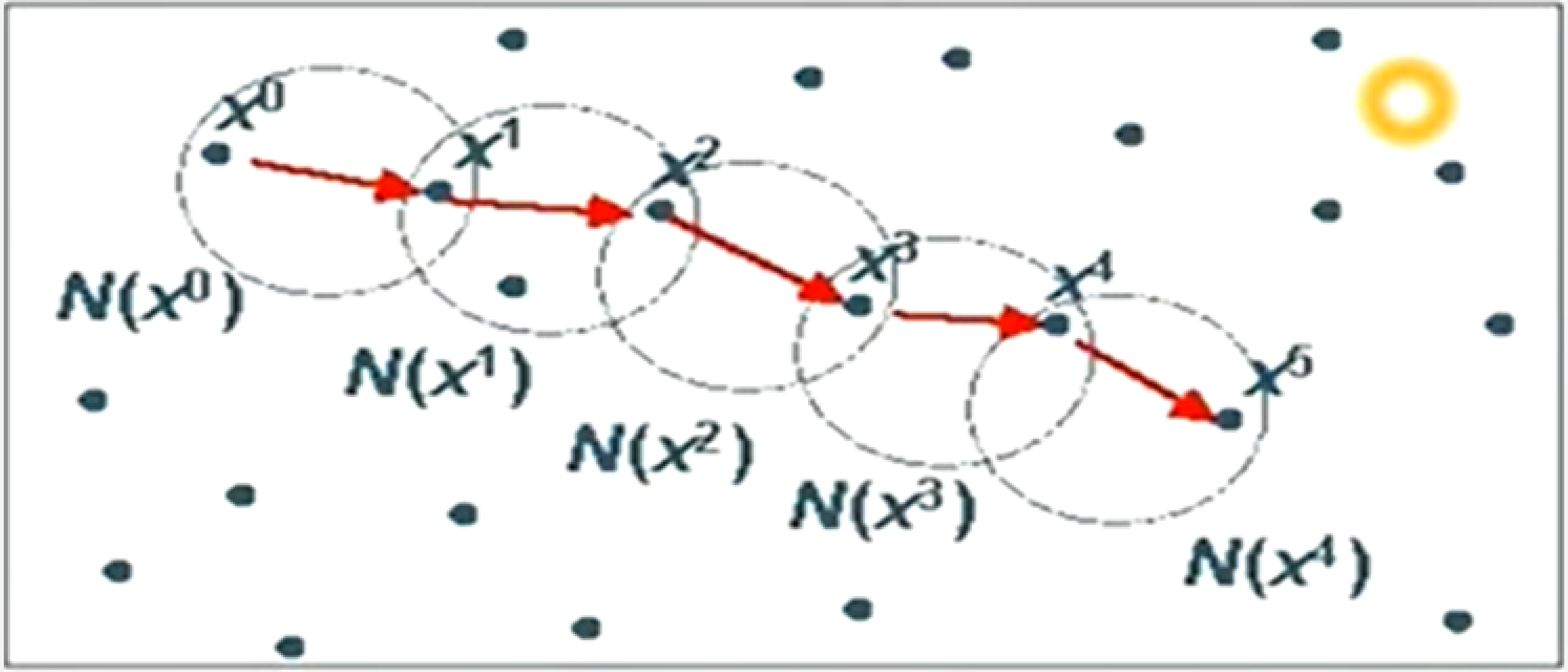
DEFINITION

LOCAL SEARCH

Local search is an algorithmic strategy used to solve optimization problems, which means problems where you seek **the best solution** from a set of candidate solutions.

Local search involves moving from one solution to another nearby solution in the space of candidate solutions (the search space or feasible space) until an optimal solution is found or a predefined stopping criterion is met.

The choice of which solution to move to is made using only information about neighboring solutions to the current solution, hence the term '**local search**.



X: solution
N(X): Set of all neighbors of solution X

Note: Local search algorithms are **suboptimal** algorithms because the search can stop even when the best solution found by the algorithm is not the best in the search space (Optimal Solution).

DESCENT METHODS

Descent methods are a specific type of local search algorithm used to find the local minimum (or maximum) of an objective function.
Descent methods are designed to improve the current solution iteratively by moving in the direction that reduces (or increases) the objective function value, depending on whether it's a minimization or maximization problem.

There are several typres of descent methods:
Gradient Descent
Steepest Descent
Subgradient Descent
...

The key distinction is that "descent methods" are a subset of "local search" algorithms.

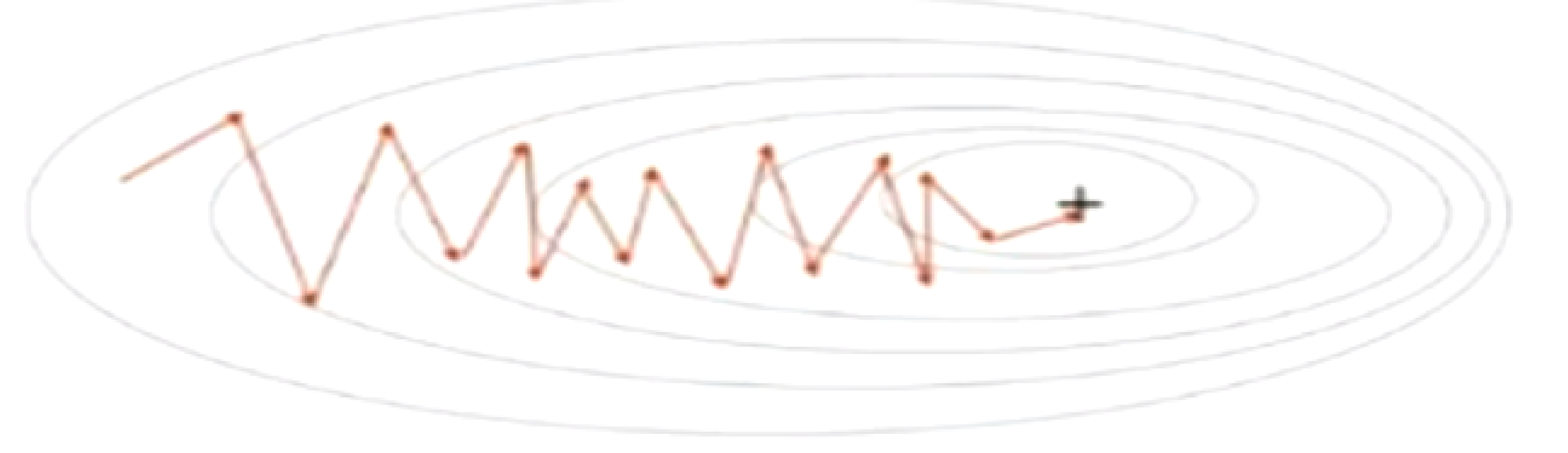
INSPIRATIONS

hill-climbing: :

the local search, was inspired by the hill-climbing algorithm. It shares the same principle of gradually moving towards an optimal solution by continuously selecting better neighboring solutions.

Memory Mechanisms:

Many local search descendants incorporate memory mechanisms, such as tabu lists or solution archives, inspired by the idea of learning from past search experiences. These mechanisms help avoid revisiting known solutions and guide the search effectively.



PRINCIPLES

PRINCIPLE OF LOCAL SEARCH

Basic Elements

Neighborhood:

- A function that associates a subset of X with each element of X.
- The concept of local optimum relative to the neighborhood.

Evaluation Function:

- Used to measure the quality of solutions.
- It represents the initial objective if working within the feasible search space.

.

Move Strategy:

- defines how the algorithm explores the search space to potentially find better solutions.

(A) INITIAL SOLUTION SELECTION

Generate an initial solution **x0 ∈ E**.
The choice can be **random** or **Constructive heuristic algorithms** like the greedy method can be used.
Note: The quality of the final solution depends on the choice of the initial solution **(a good start allows for a quicker convergence to the final solution)**.

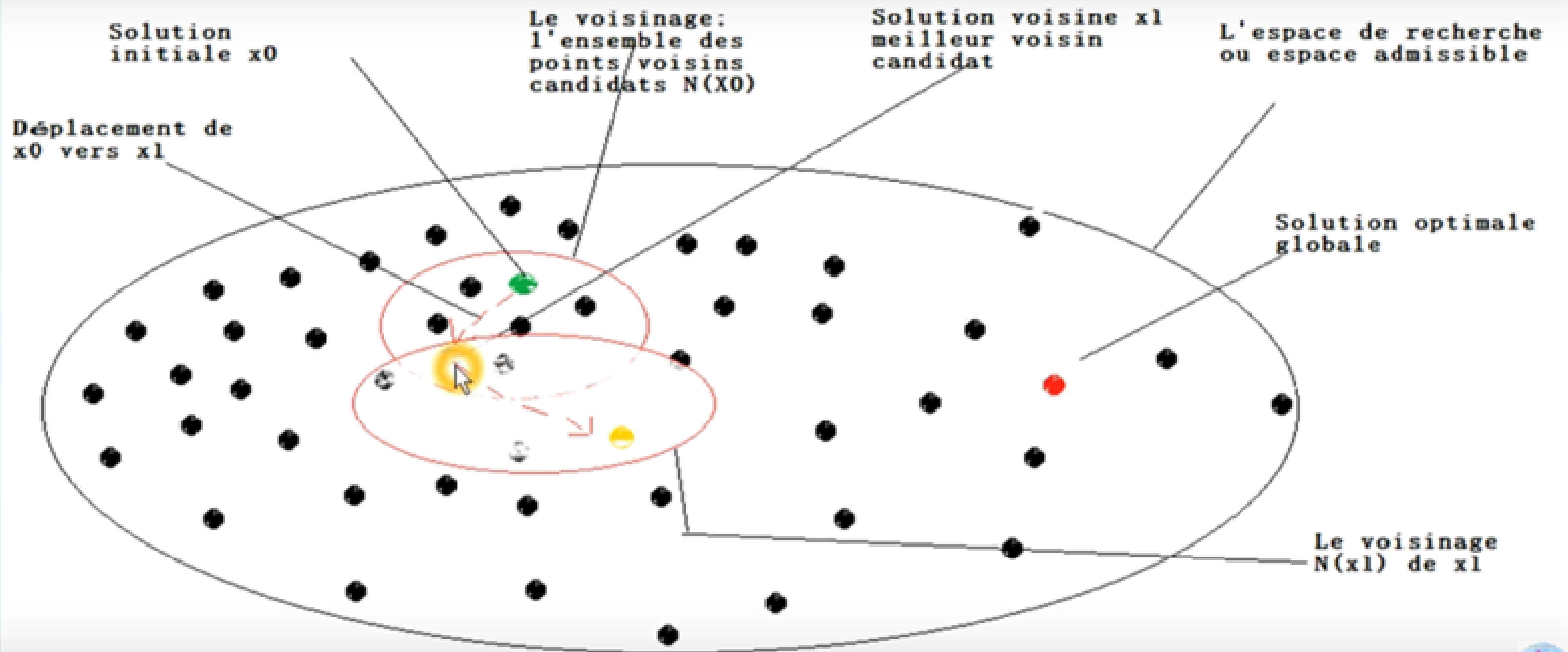
(B) CHOICE OF TERMINATION CRITERIA

Select a stopping criterion to prevent a large number of iterations or cases of looping or blocking.
The termination criterion for local search can be:
- A time limit (the time can be relative to the number of iterations, for example, as long as k = 100).
- Another option is to stop when the best solution found by the algorithm has not been improved for a specified number of iterations (k = 10)."

(C) DETERMINING THE NEIGHBOR

Generate a solution xi in the neighborhood **N(xi-1) of xi-1**.
There are several ways to choose a neighbor:

1. Randomly selecting a neighbor from those that improve the current solution (**first improvement**) => one of the neighbors that minimizes the objective function.
2. Selecting the best neighbor that improves the current solution (**best improvement**) => iterating through all neighbors (**more costly**)



DECISION PROCESS

Decision Process

- **General descent method**
  - Data : f , E
  - Initialization of an initial solution (Randomly or with an heuristic)
  - initial solution (current) x0
  - current optimal solution x* = x0 , k = 1
  - while stop condition is not reached
    1. Generation of candidate neighbors : neighborhood
    2. Choice of the best candidate neighbor xk
    3. if f(xk) < f(x*) , so x* = xk , k = k + 1

- **Gradiant descent method**

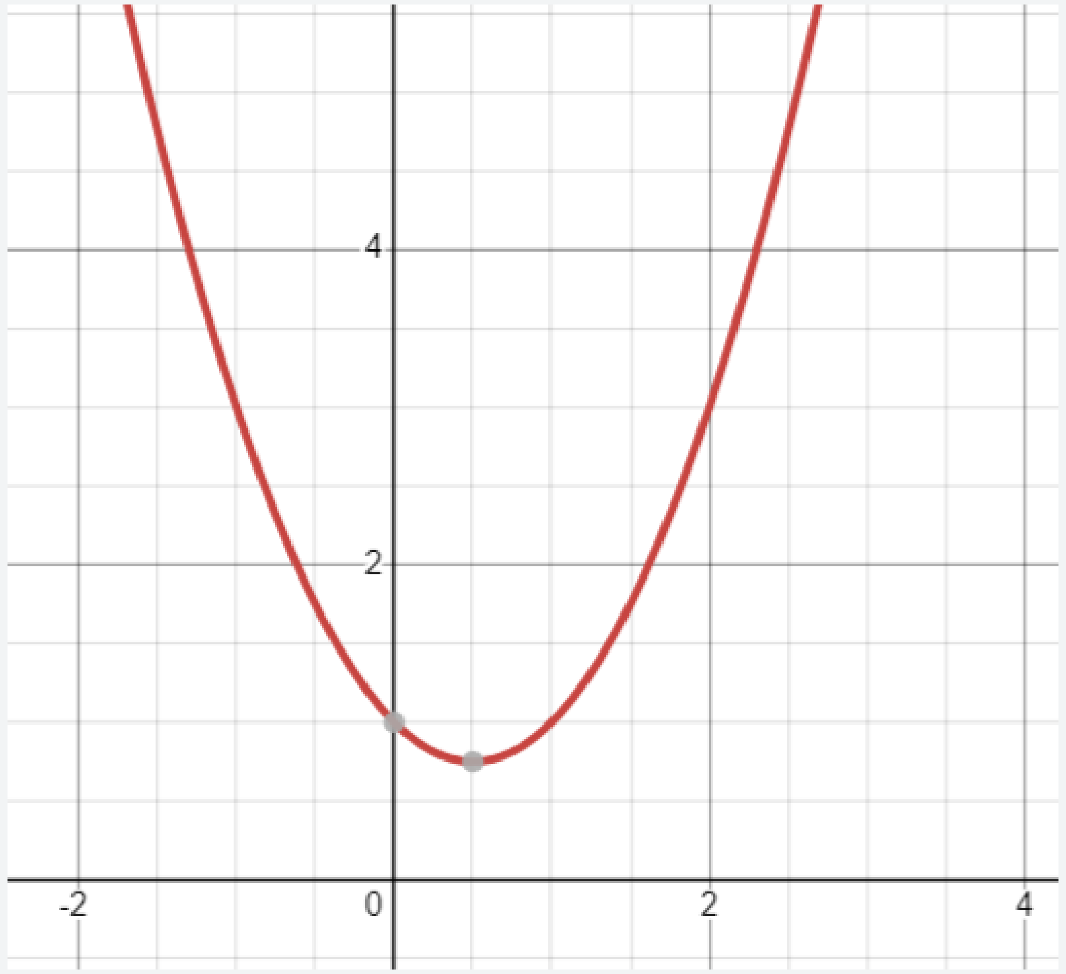
To determine the neighboring solution, the gradient descent method uses

- The gradiant to calculate the direction
- A step for moving towards the neighboring solution (Hyperparameter)
- This a convexe function with one global minimum

$f : R^{(n)} \longrightarrow R$

$x \in R^{(n)}$: sloution vector

$\alpha \in [0,1]$



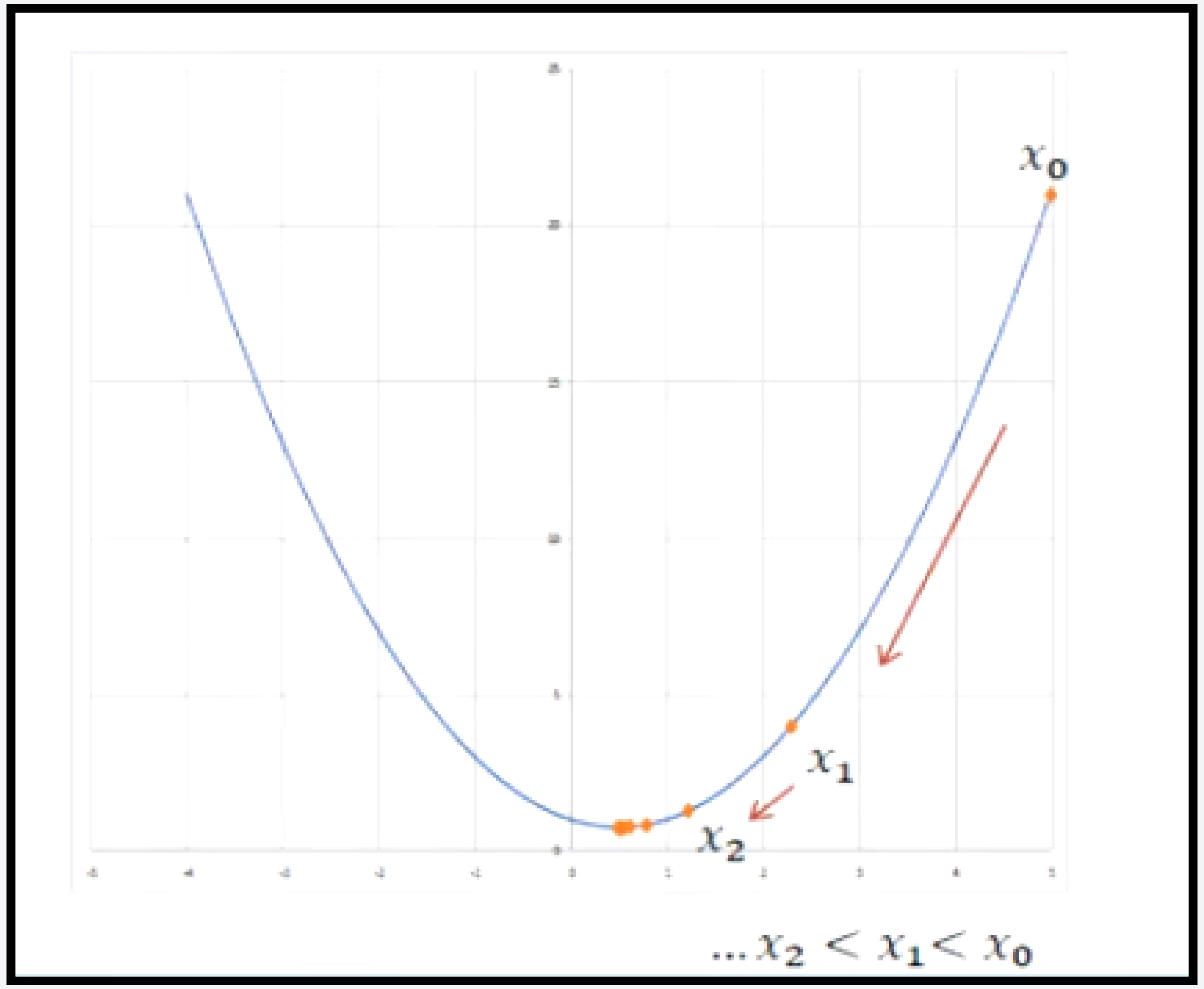
- The derivative provides information about the slope of the function

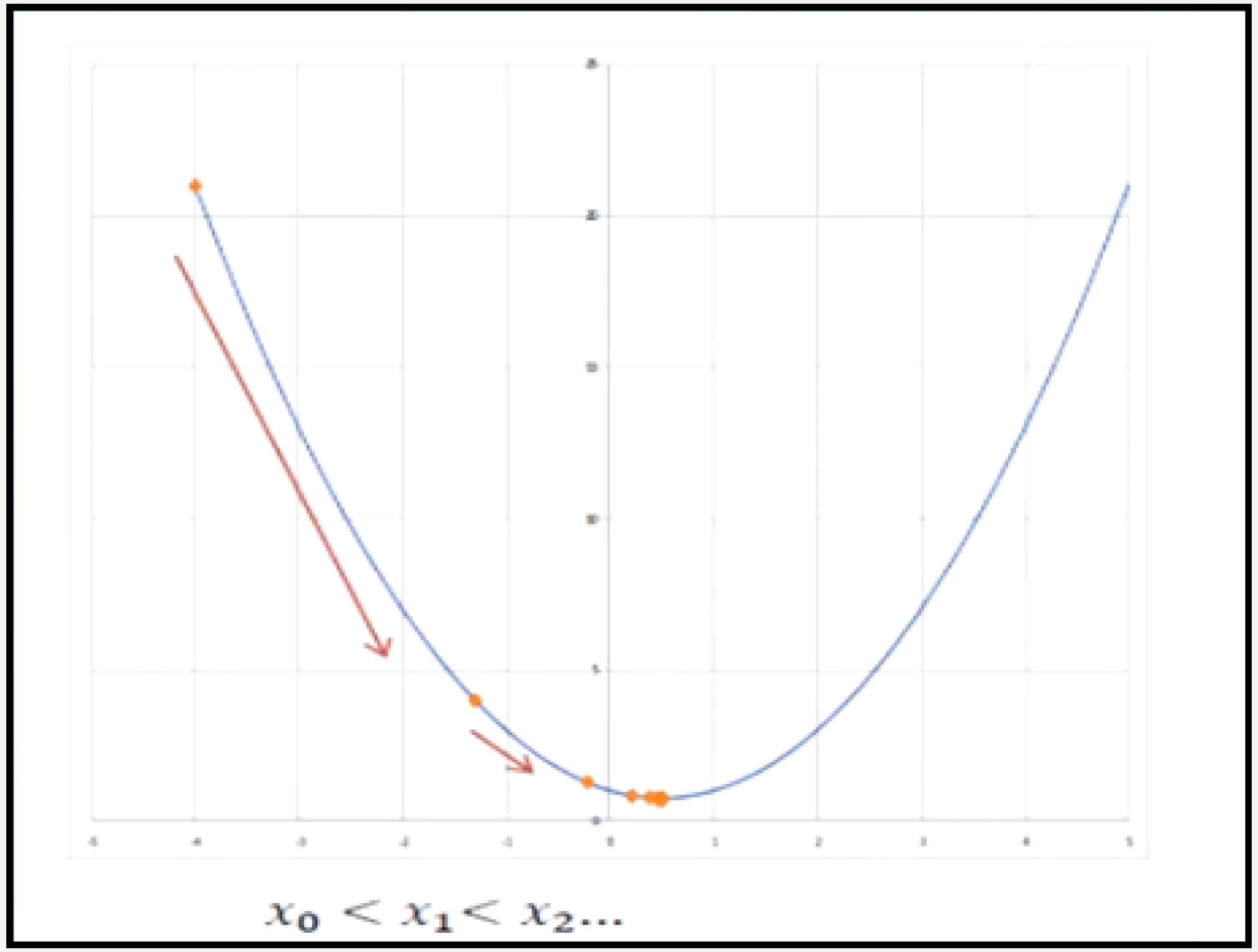
- if $\nabla f(x) > 0$, **so f is increasing in the direction of d**

- if $\nabla f(x) < 0$, **so f is decreasing in the direction of d**

- The derivative d provides information about the slope of the function

$$x_{n+1} = x_n - \alpha \nabla f(x_n)$$





ALGORITHMS & ILLUSTRATIONS

Local SEARCH:

Start:

Initial solution obtained by a constructive method or randomly

Iteration:

Improvement of the current solution by searching within its neighborhood

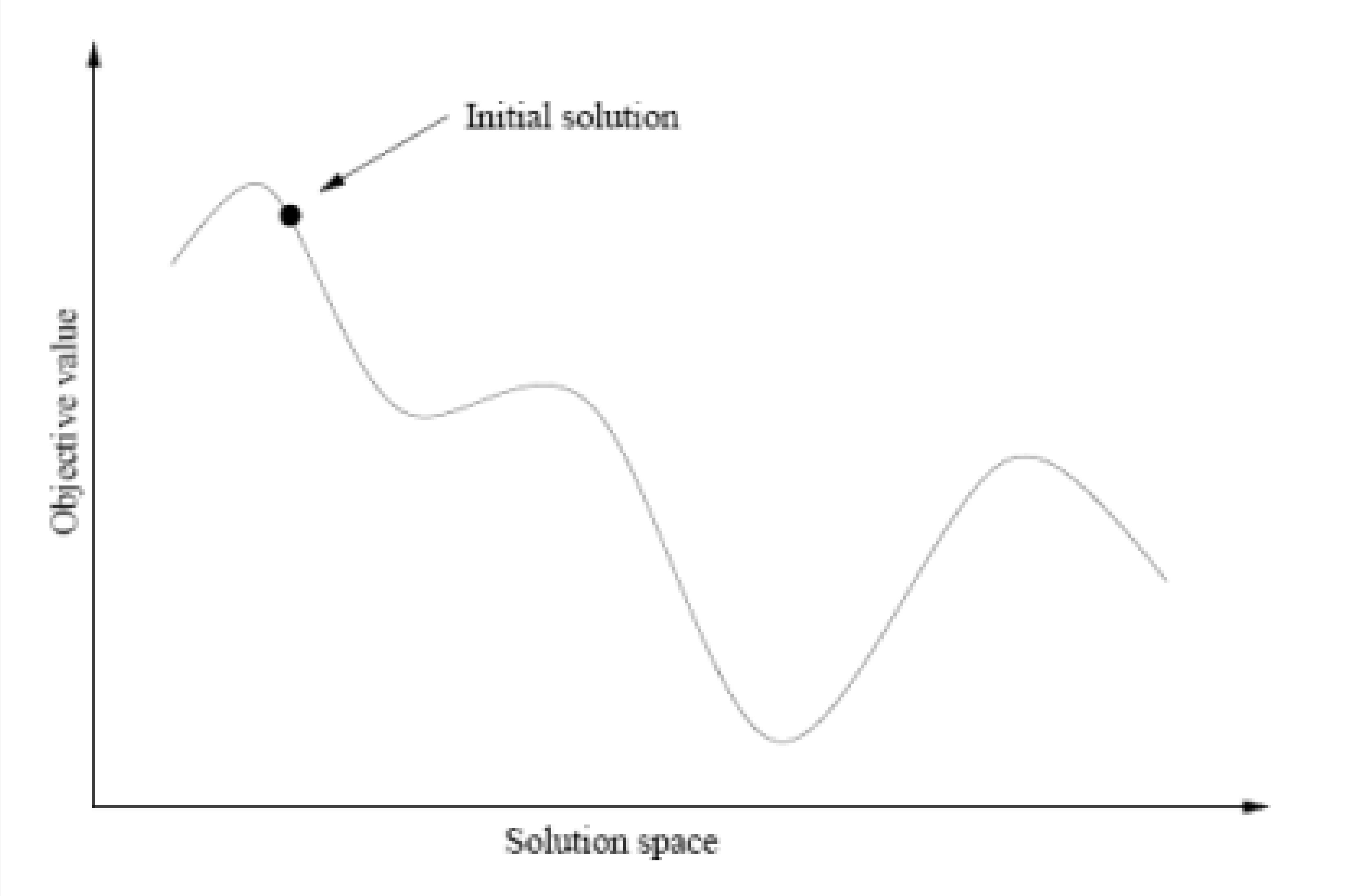
Termination:

First local optimum found (there is no better neighboring solution than the current solution)

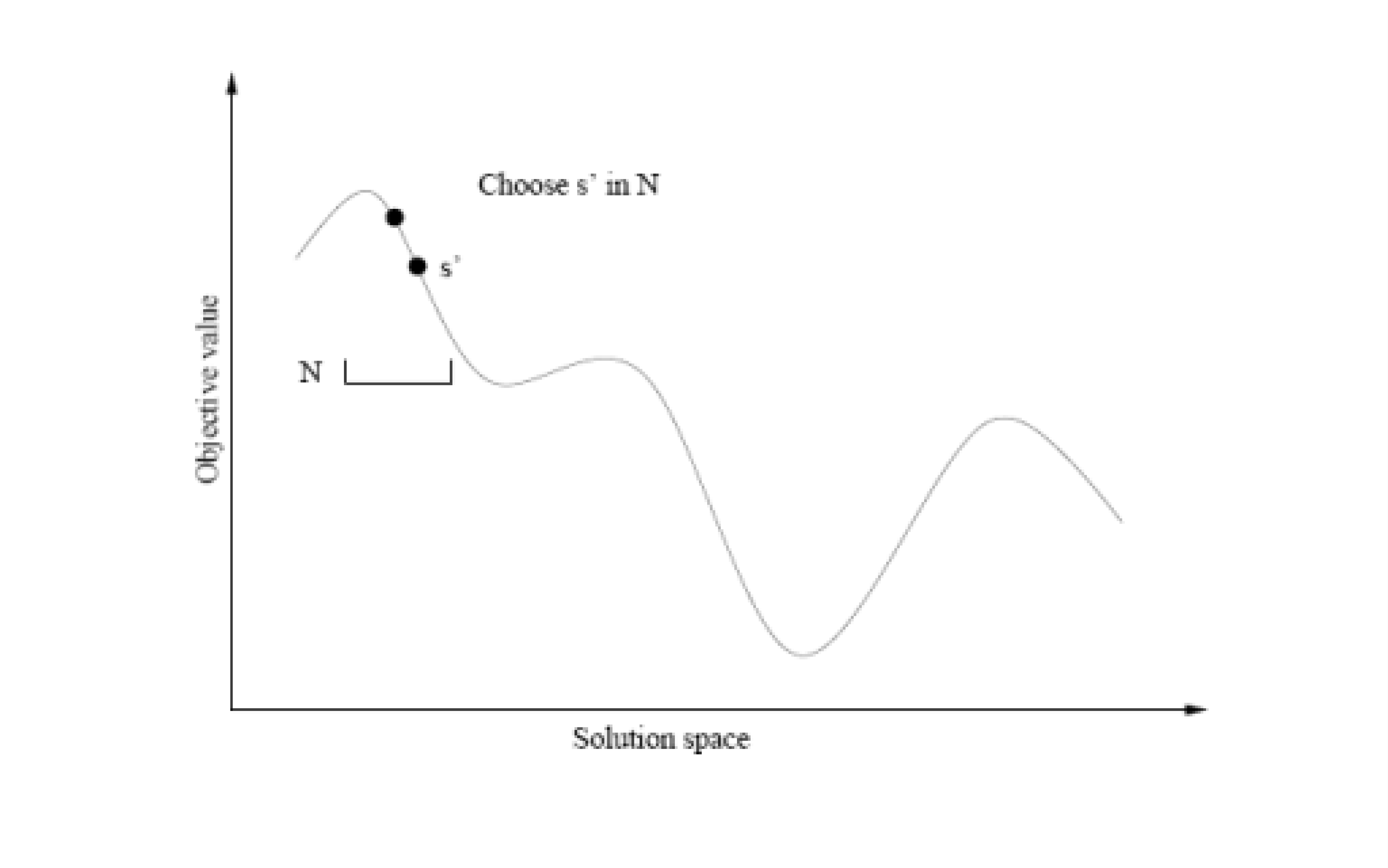
local search algorithme :

```
Algorithme localSearch ( sO)
      s  ← sO
     arrêt ← false
     while arrêt = false do
          generate a neighborhood N(s)
          #there doesn't exist an s' in N(s)
               if f(s') > f(s) then arrêt ← true
           else
                choose a neighbor s' of s while
                     f(s') < f(s)
                     s ← s'
           endif
      endwhile
      return s
endAlgo
```

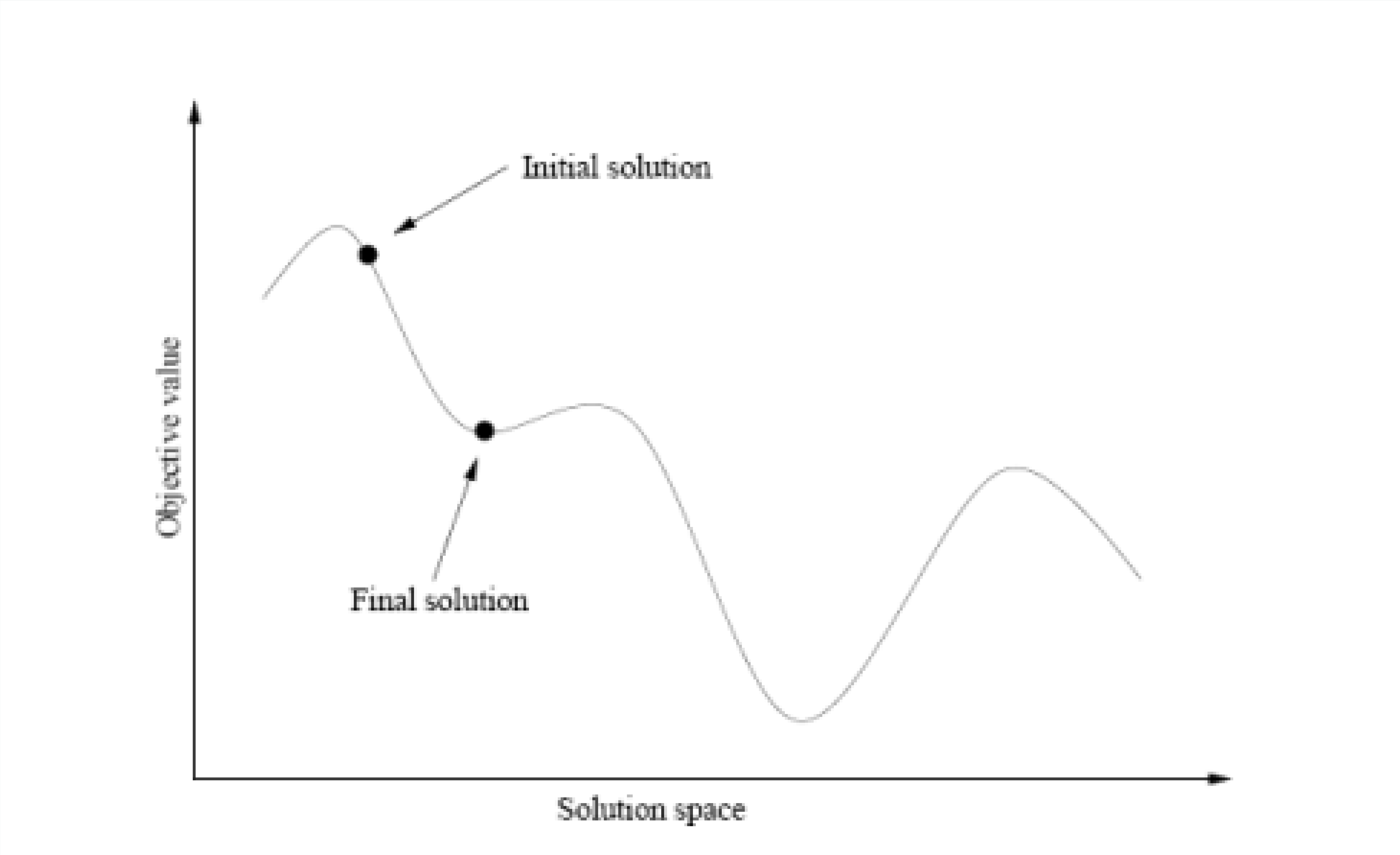
local search simple Illustration :



local search simple Illustration :



local search simple Illustration :



ADVANTAGES & LIMITATIONS

Advantages :

- **Convergence** : Descent methods often converge to a local minimum (or maximum) of the objective function, making them suitable for a wide range of optimization problems.
- **Simplicity** : Many descent methods are straightforward to implement and require minimal computational resources.
- **Scalability** : These methods can handle large datasets and high-dimensional parameter spaces effectively

Limites :

- **Local Minima** : Descent methods can get trapped in local minima or maxima, failing to find the global minimum or maximum
- **Sensitivity to Initialization** : The performance of descent methods can be highly sensitive to the initial values of parameters
- **Hyperparameter Sensitivity :** Descent methods often involve hyperparameters (learning rate) that require careful tuning to achieve good results

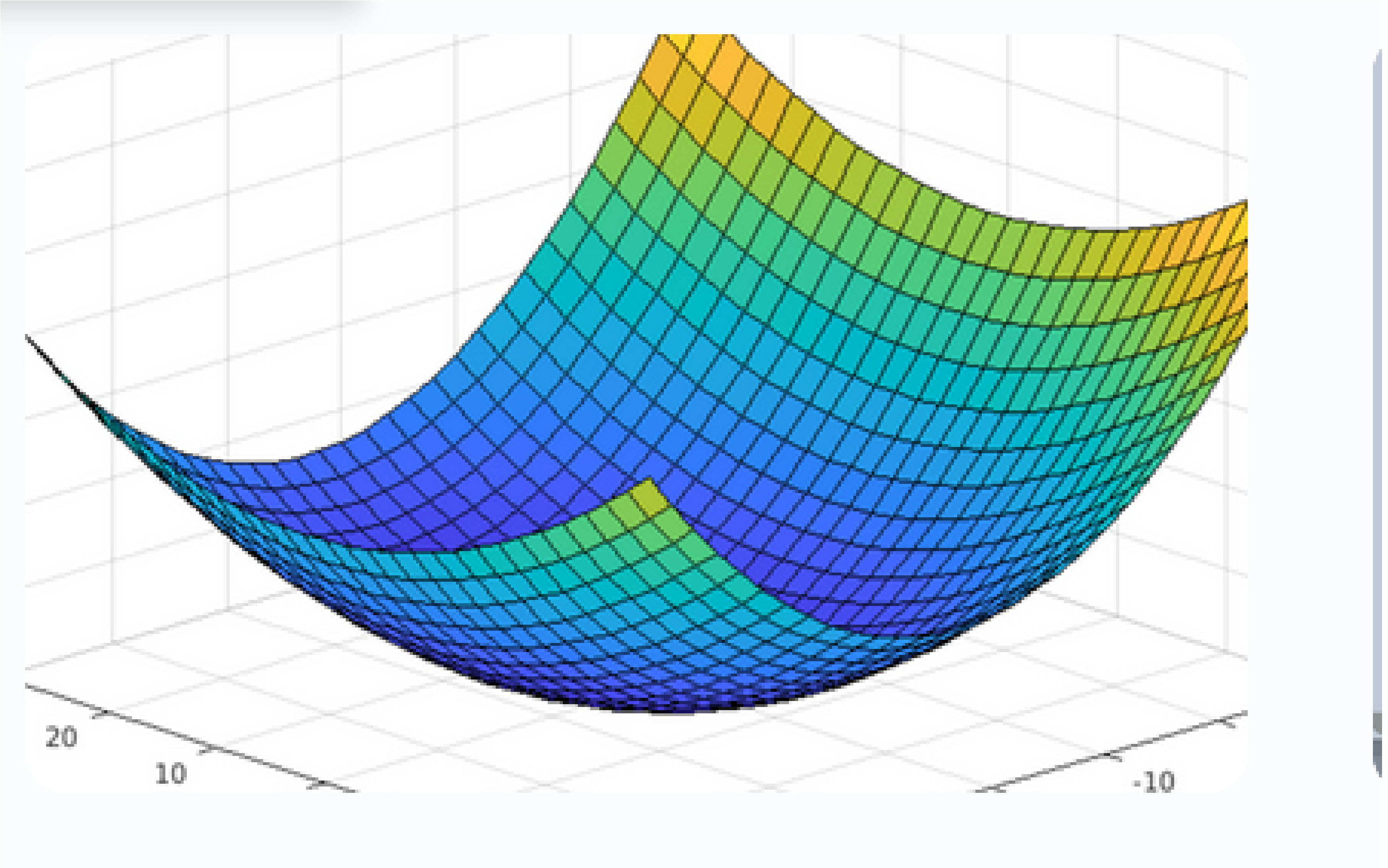
APPLICATION

APPLICATIONS OF THE DESCENT METHOD

The Descent Method has a wide range of applications in different fields. Some of the common applications include:

Machine Learning and Deep learning

Gradient Descent, a type of descent method, is crucial for training machine learning models, such as neural networks, to minimize loss functions.

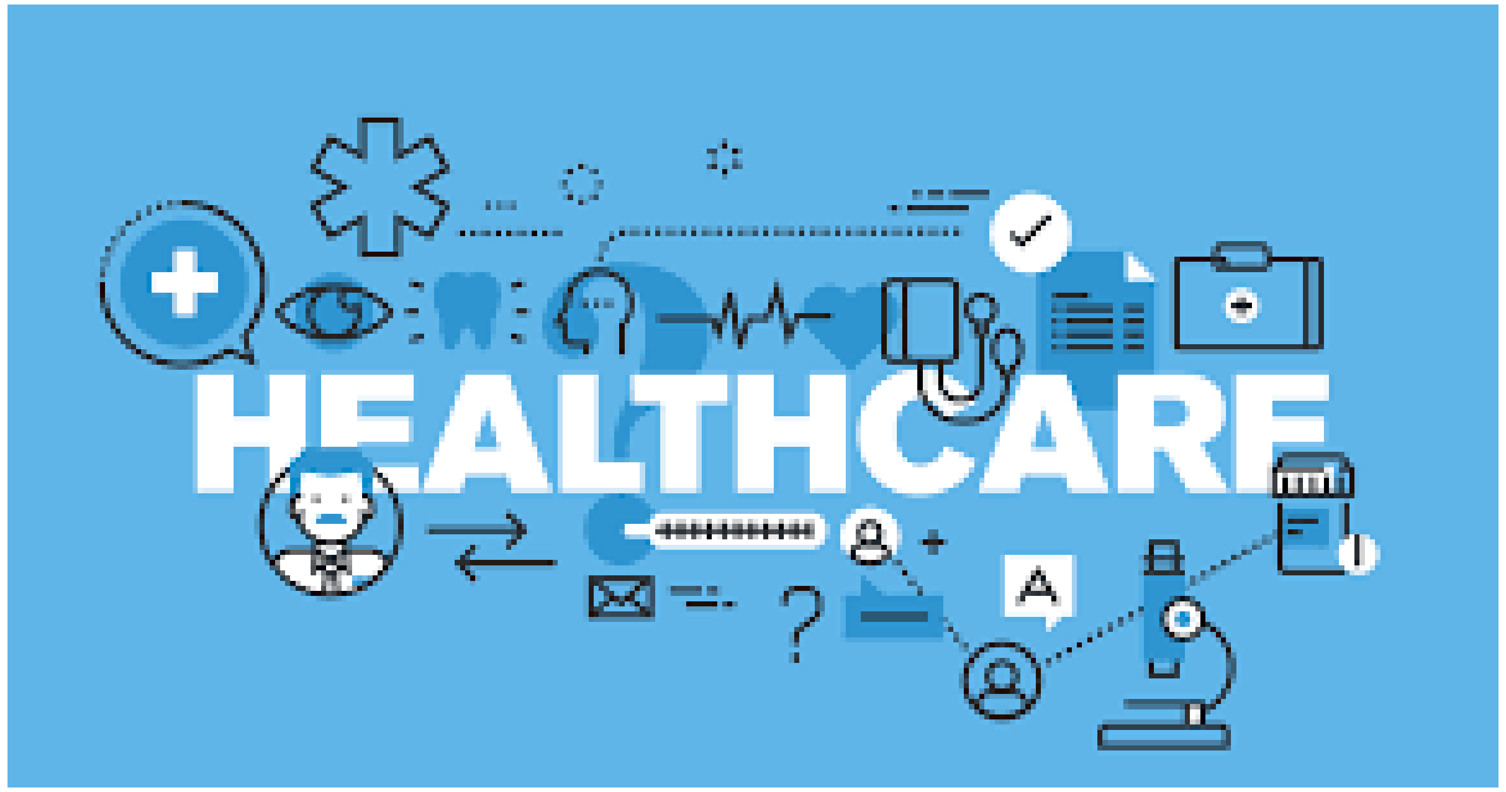


Economics

They play a role in economic modeling, game theory, and optimization of economic systems.

Healthcare

Optimization techniques are applied in medical image registration, treatment planning, and epidemiological modeling.



IMPLEMENTATION

implementation of descent method "downhill method":

Data: $f, E,$

Initialization: Choose an initial solution (randomly or with a heuristic)

Initial solution (current) $x_0,$

Current optimal solution $x^* = x_0, k = 1$

While the termination criterion is not met:

1. Generating candidate neighbors: neighborhood
2. Choosing the best candidate neighbor x_k
3. If $f(x_k) < f(x^*)$ then $x^* = x_k, k = k + 1$, Go to (1)

   End if

   End while

- Both the traditional local search method and the descent method focus on improving solutions step by step. However, the key difference lies in how they choose the next solution to consider:

- **Traditional Local Search:** It explores neighboring solutions, even if they might be worse than the current solution, in the hope of finding a better solution in the long run.

- **Descent Method:** This method only considers neighboring solutions that are better than the current one. It stops when it can't find any better neighboring solution.

PROBLEM

The Traveling Salesman Problem (TSP) is a classic optimization problem where the goal is to find the shortest possible route that visits a set of cities and returns to the starting city

- **Initial Solution**: Start with an initial solution. This can be a random tour, a heuristic solution, or any other method to generate an initial feasible solution.
- **Evaluation:** Calculate the total distance or cost of the current tour. This is the objective function you want to minimize
- **Neighborhood Generation:** Define a neighborhood structure. In the Descent method, this means defining how to generate neighboring solutions. Common approaches include swapping two cities, reversing a segment of the tour, or inserting a city into a different position in the tour.

- **Local Search:** Choose a neighboring solution and apply the move to get a new solution. Evaluate the new solution.

- **Improvement:** If the new solution is better (i.e., it has a shorter total distance) than the current solution, accept it as the new current solution.

- **Termination Criteria:** Define stopping conditions. stop when a time limit is reached, when no further improvement is possible, or after a certain number of iterations.

THANK YOU !!!!