CIS
RESEARCH GROUP

Heuristics & Metaheuristics for Optimization & Learning

Mario Pavone

mpavone@dmi.unict.it

http://www.dmi.unict.it/mpavone/

AA. 2024/2025 - LM 18

Solving Problem Methods

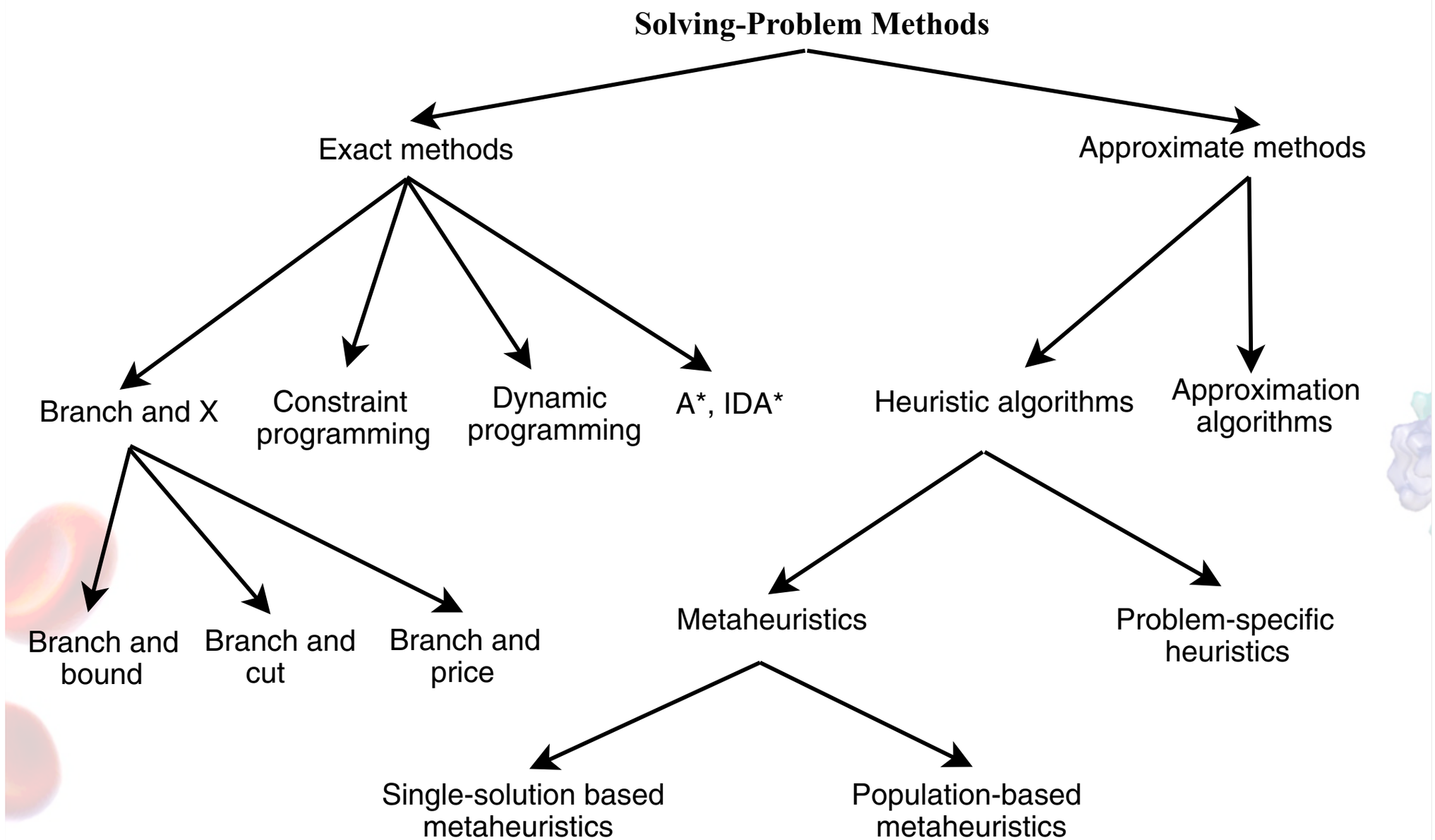


FIGURE: Classical Solving-Problem Methods

Key Concepts to Pay Attention

- Codifica del problema
- Rappresentazione della soluzione
- Progettazione metodo/processo di ricerca della soluzione
- Definizione della funzione obiettivo

GREEDY ALGORITHMS

In Greedy Algorithm, we start from scratch (empty solution)

Algorithm 1.2 Template of a greedy algorithm.

$s = \{\}$; /* Initial solution (null) */
Repeat
e_i = Local-Heuristic($E \backslash \{e / e \in s\}$) ;
/* next element selected from the set E minus already selected elements */
If $s \cup e_i \in F$ **Then** /* test the feasibility of the solution */
$s = s \cup e_i$;
Until Complete solution found

Popular techniques as they are simple to design

GREEDY ALGORITHMS

In Greedy Algorithm, we start from scratch (empty solution)

Algorithm 1.2 Template of a greedy algorithm.

There is no BackTracking of already taken decisions

$s = s \cup e_i$;
Until Complete solution found

Popular techniques as they are simple to design

GREEDY ALGORITHMS: LIMITS

In most of solving problems the local view of greedy heuristics decreases their performance compared to iterative algorithms

At each step a heuristic is used to select the next element to be part of the solution: chooses the best element in terms of its contribution in optimizing locally the objective function

Local Optimality does not implicate a global optimality

Greedy heuristics are in general myopic in their construction of a solution

GREEDY ALGORITHM: AN EXAMPLE

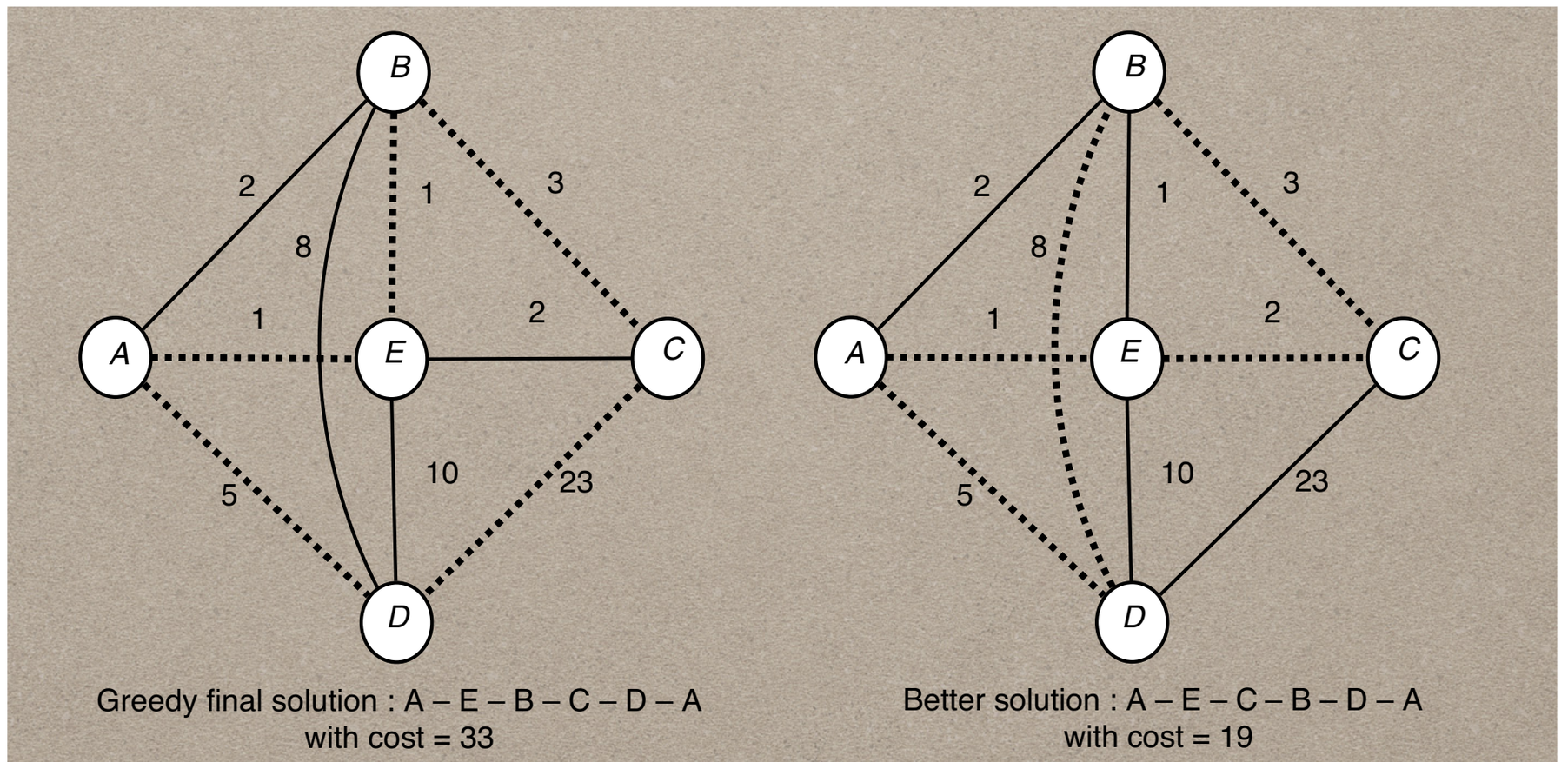


FIGURE 1.10 Illustrating a greedy algorithm for the TSP using a static heuristic. An element is associated with an edge of the graph, and the local heuristic consists in choosing the nearest neighbor. The obtained solution is $(A - E - B - C - D - A)$ with a total cost of 33, whereas a better solution with a cost of 19 is given (right).

ITERATIVE METHODS

iteratively apply the solutions generation, and replacement procedures from the current single solution

How?

local transformations

Which means....

"walks" through neighborhoods

Until?

a given stopping criteria is reached

ITERATIVE METHODS

Walks

=

Generation phase

from the current solution

+

Local transformations

+

Replacement phase

New solution

ITERATIVE METHODS

Algorithm 2.1 High-level template of S-metaheuristics.

Input: Initial solution s_0.
$t = 0$;
Repeat
  /* Generate candidate solutions (partial or complete neighborhood) from s_t */
  Generate($C(s_t)$) ;
  /* Select a solution from $C(s)$ to replace the current solution s_t */
  s_{t+1} = Select($C(s_t)$) ;
  $t = t + 1$;
Until Stopping criteria satisfied
Output: Best solution found.

ITERATIVE METHODS

Algorithm 2.1 High-level template of S-metaheuristics.

Input: Initial solution s_0.
$t = 0$;
Repeat
  /* Generate candidate solutions (partial or complete neighborhood) from s_t */
  Generate($C(s_t)$) ;
  /* Select a solution from $C(s)$ to replace the current solution s_t */
  s_{t+1} = Select($C(s_t)$) ;
  $t = t + 1$;
Until Stopping criteria satisfied
Output: Best solution found.

generation

replacement

The common search concepts are the definition of the ***neighborhood*** structure and the determination of the ***initial solution***

ITERATIVE METHODS

The neighborhood structure plays a crucial role in the performance of an algorithm

$i = 0;$

Repeat

generation

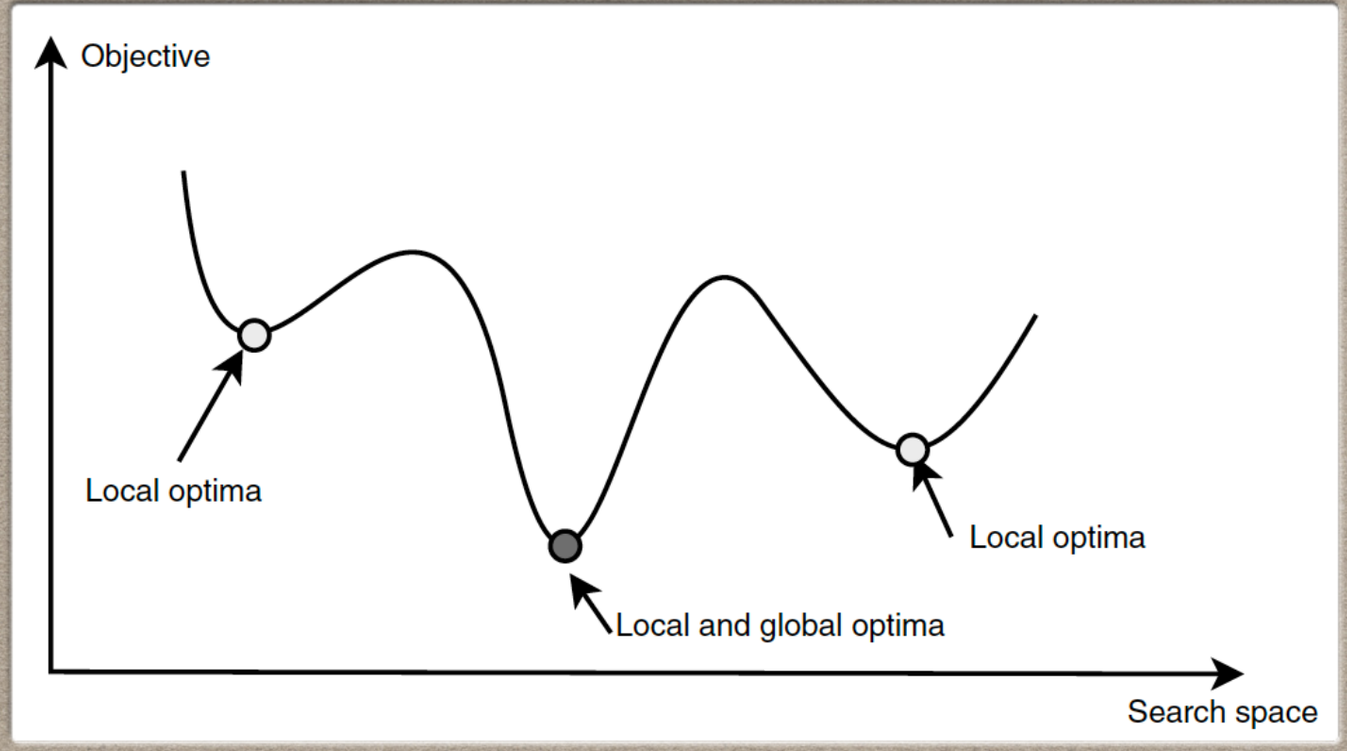
replacement

If the neighborhood structure is not adequate to the problem, any algorithm will fail to solve the problem

The common search concepts are the definition of the ***neighborhood*** structure and the determination of the ***initial solution***

LOCAL OPTIMA DEFINITION

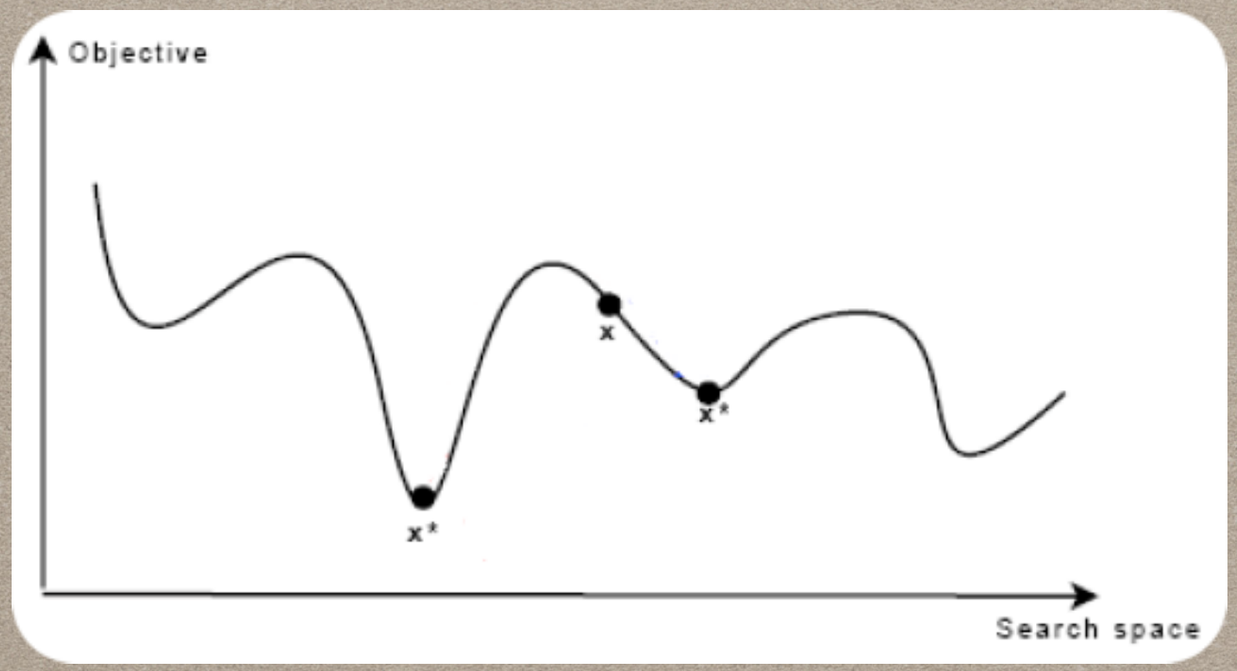
Definition 2.4 Local optimum. *Relatively to a given neighboring function N, a solution $s \in S$ is a local optimum if it has a better quality than all its neighbors; that is, $f(s) \leq f(s')^2$ for all $s' \in N(s)$ (Fig. 2.4).*



NOTE: for the same problem, a local optimum for a neighborhood N_1 may not be a local optimum for a different neighborhood N_2

INITIAL SOLUTION

Choosing the initial solution is crucial



The quality of the local optima obtained by the considered solving method depends on the Initial Solution

ITERATIVE METHODS

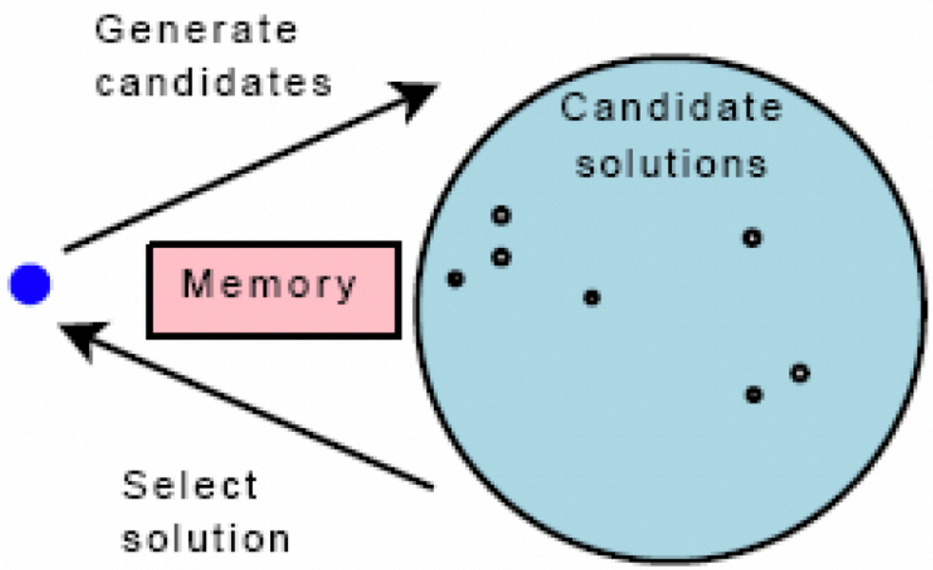


Fig. 2.1 Main principles of single-based metaheuristics.

[E.G. Talbi, Metaheuristics: from design to implementation, Wiley 2009]

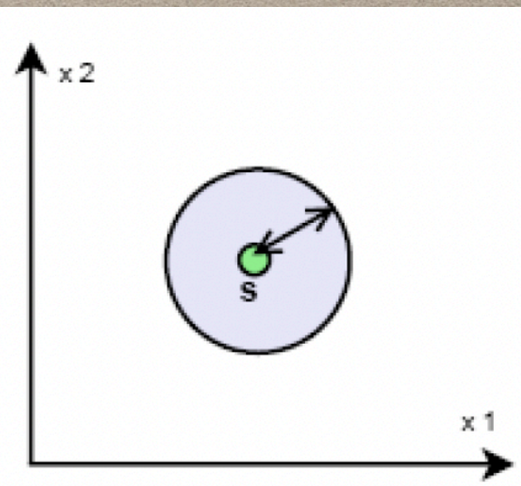
[E.G. Talbi, Metaheuristics: from design to implementation, Wiley 2009]

NEIGHBORHOOD

Formally:

Definition 2.1 Neighborhood. *A neighborhood function N is a mapping $N : S \rightarrow 2^S$ that assigns to each solution s of S a set of solutions $N(s) \subset S$.*

Fundamental property: locality – The effect on the solution (phenotype) when performing a move (small perturbation) to the representation (genotype): Strong locality



The circle represents the neighborhood of s in a continous problem with 2 dimensions.

Locality: the effect on the solution when performing the perturbation (move) in the representation

When small changes are made in the representation, the solution must reveal small changes

NEIGHBORHOOD

The neighborhood definition depends strongly on the representation associated with the problem at hand.

The natural neighborhood for **binary representations** is based on the **Hamming Distance**

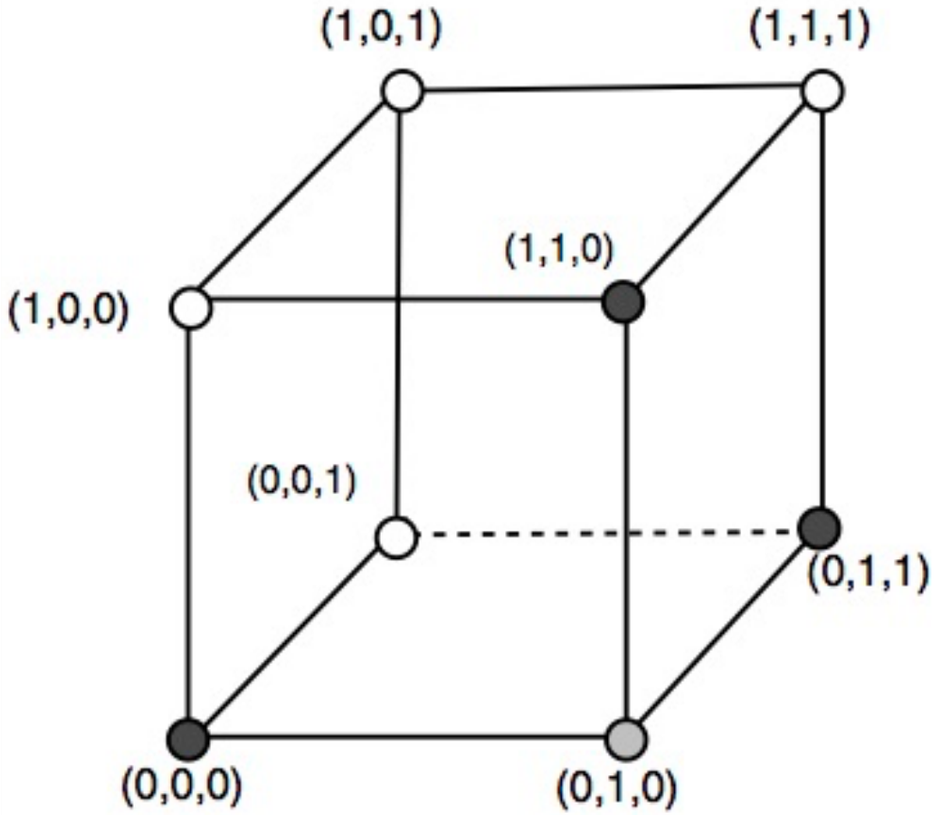
For a **binary vector of size *n***, the **size of the neighborhood will be *n***

For a **permutation of size *n***, the **size of the neighborhood is *n(n-1)/2***

A usual neighborhood is based on the **swap operator**

[E.G. Talbi, Metaheuristics: from design to implementation, Wiley 2009]

NEIGHBORHOOD



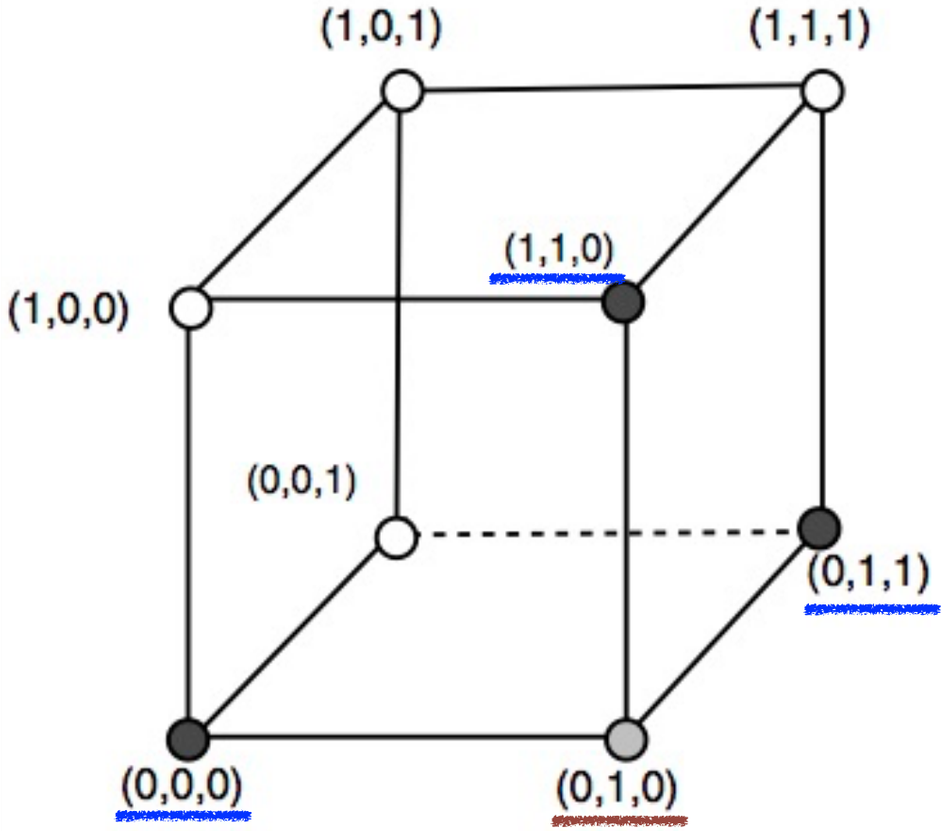
- Nodes of the hypercube represent solutions of the problem.
- The neighbors of a solution (e.g., (0,1,0)) are the adjacent nodes in the graph.

Neighborhoods for a discrete binary problem

Hamming distance

The neighborhood of a solution s consists in flipping one bit of the solution.

NEIGHBORHOOD



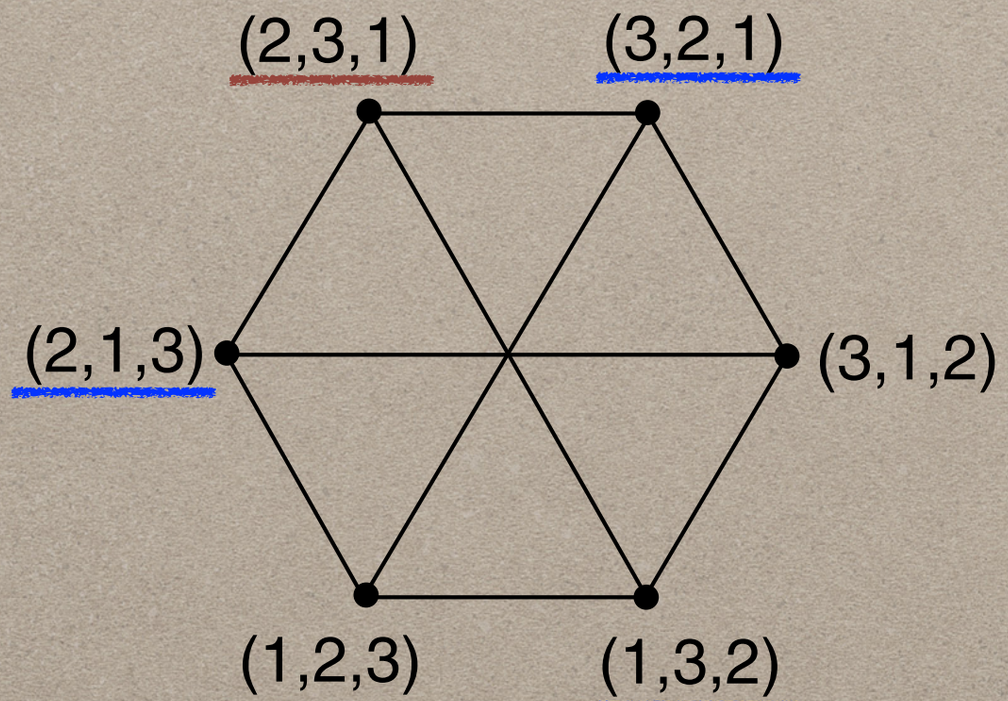
- Nodes of the hypercube represent solutions of the problem.
- The neighbors of a solution (e.g., (0,1,0)) are the adjacent nodes in the graph.

Neighborhoods for a discrete binary problem

Hamming distance

The neighborhood of a solution s consists in flipping one bit of the solution.

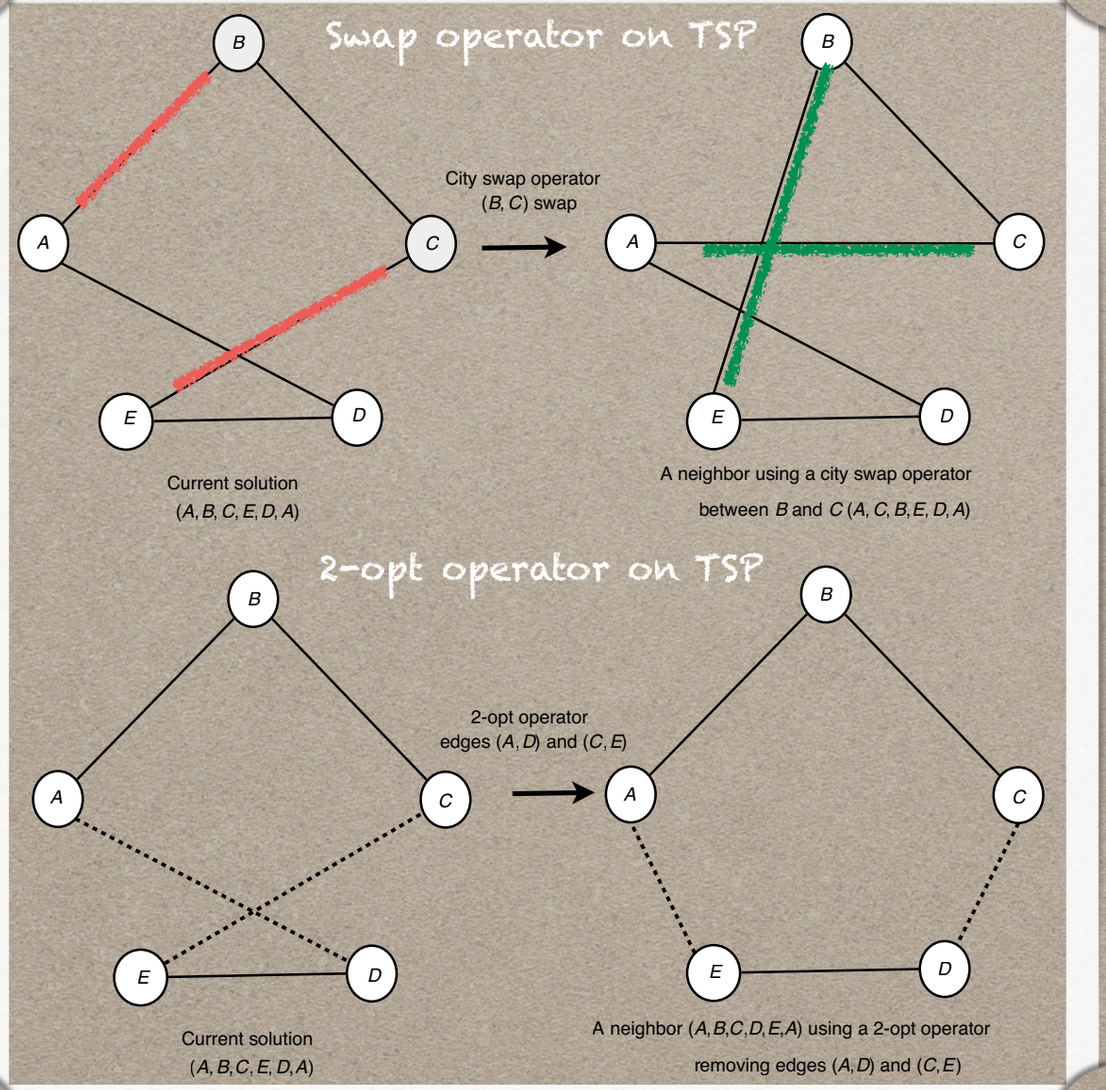
NEIGHBORHOOD

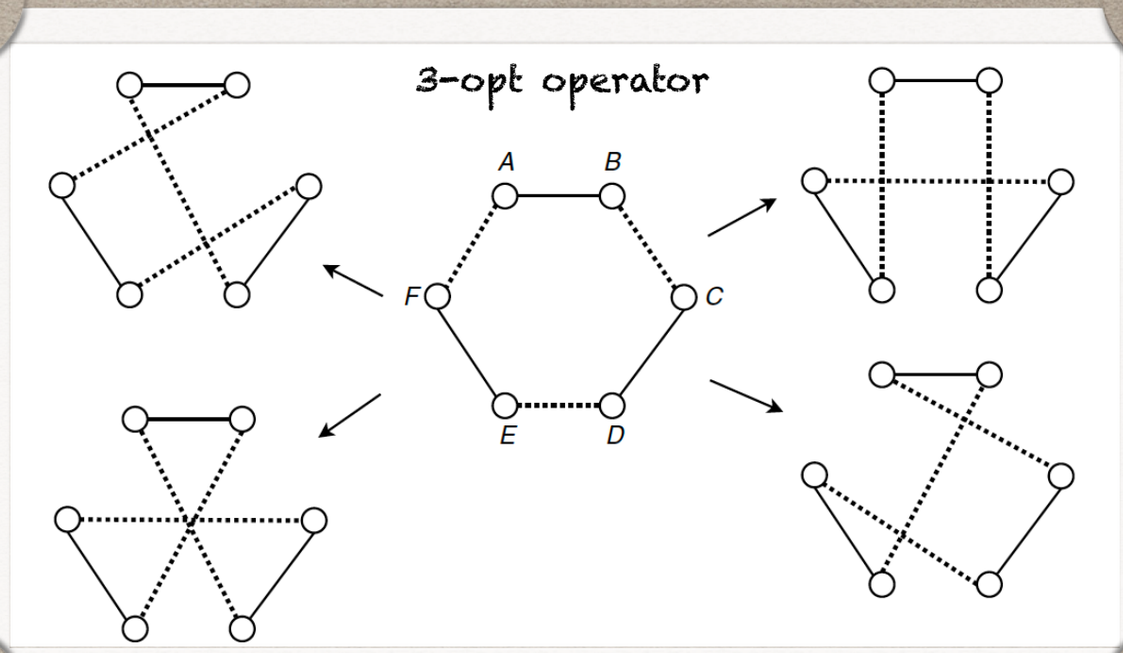


Permutations-based representation:
Swap Operator

An example of neighborhood for permutation problem of size 3

NEIGHBORHOOD





K-opt operator: k edges are removed from the solution and replaced with other k edges

The neighborhood for the 2-opt operator is represented by all the permutations obtained by removing two edges

NEIGHBORHOOD

It is important to highlight that the efficiency of a neighborhood is related not only to the representation but also to the type of problems to solve

- Scheduling problems: permutation represents a priority queue. The relative order in the sequence is important (2-opt → weak locality).
- Routing problems: adjacency of the element is important. (2-opt → strong locality). (ex. TSP)

NEIGHBORHOOD

It is important to highlight that the

NOTE: for scheduling problems the k-opt family of operator is not suited.

Position-Based neighborhood -> Insertion (Move) operator

Order-based neighborhood -> Swap Operator or Inversion Operator

relative order in the sequence is important (2-opt → weak locality).

- Routing problems: adjacency of the element is important. (2-opt → strong locality). (ex. TSP)

NEIGHBORHOOD

FIGURE 2.7 Insertion operator.

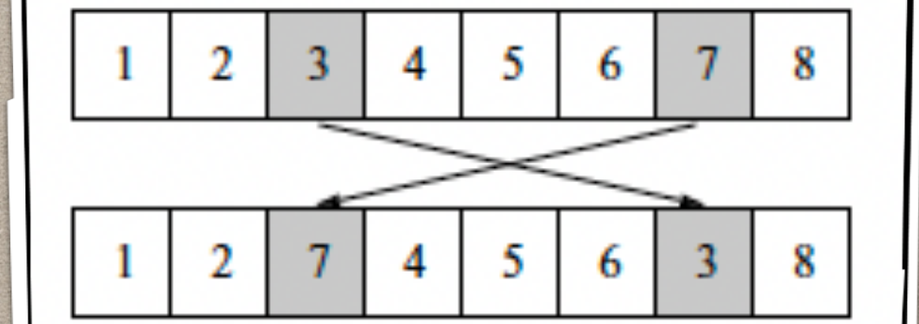


FIGURE 2.8 Exchange operator.

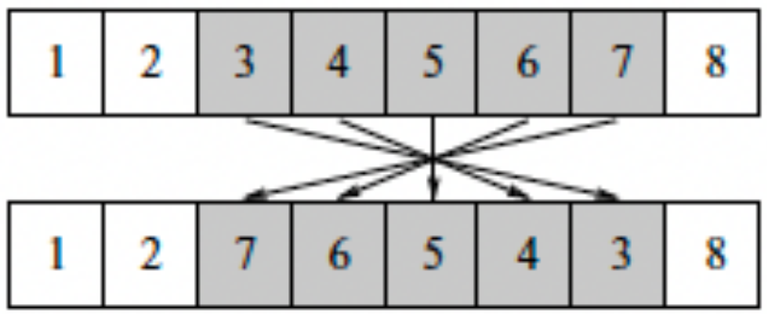
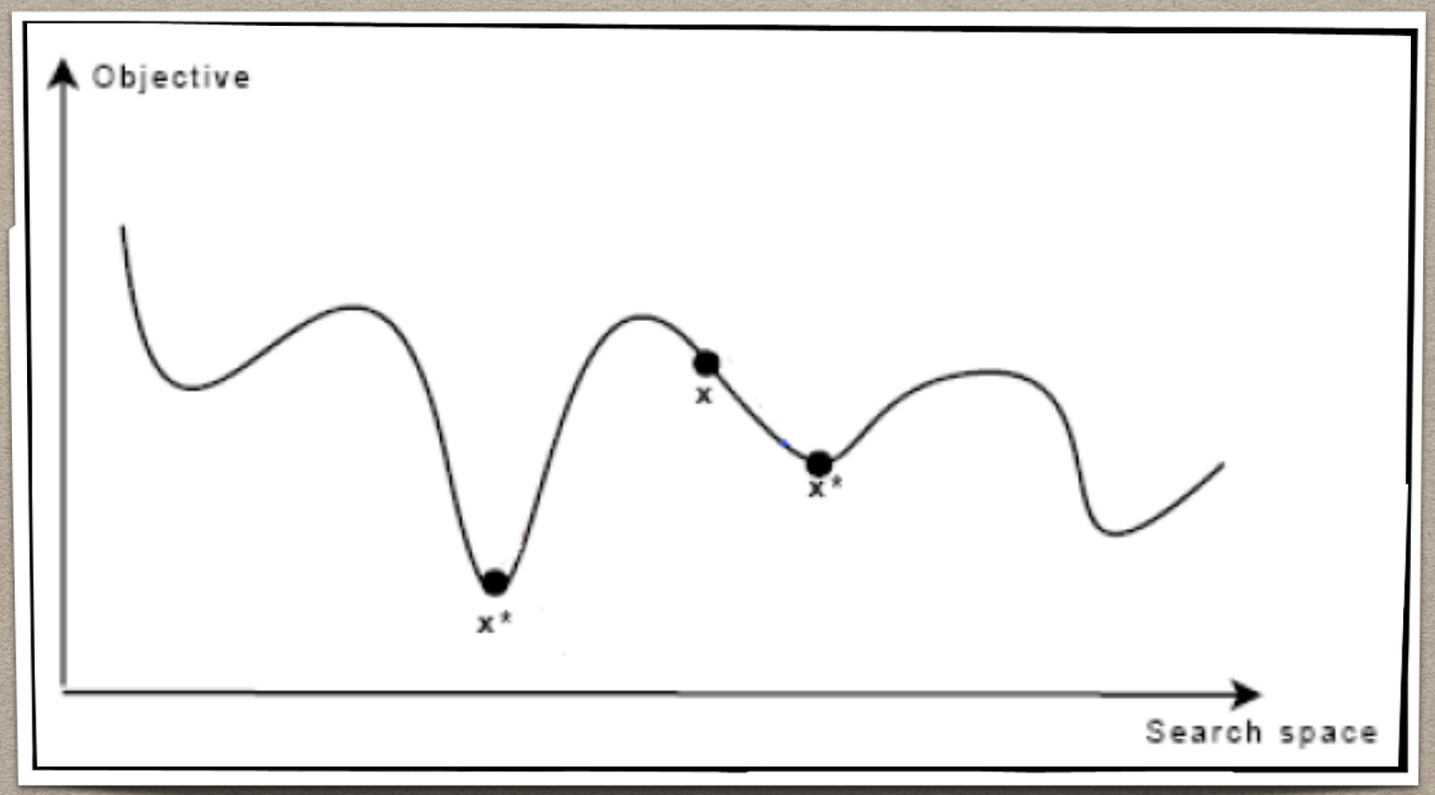
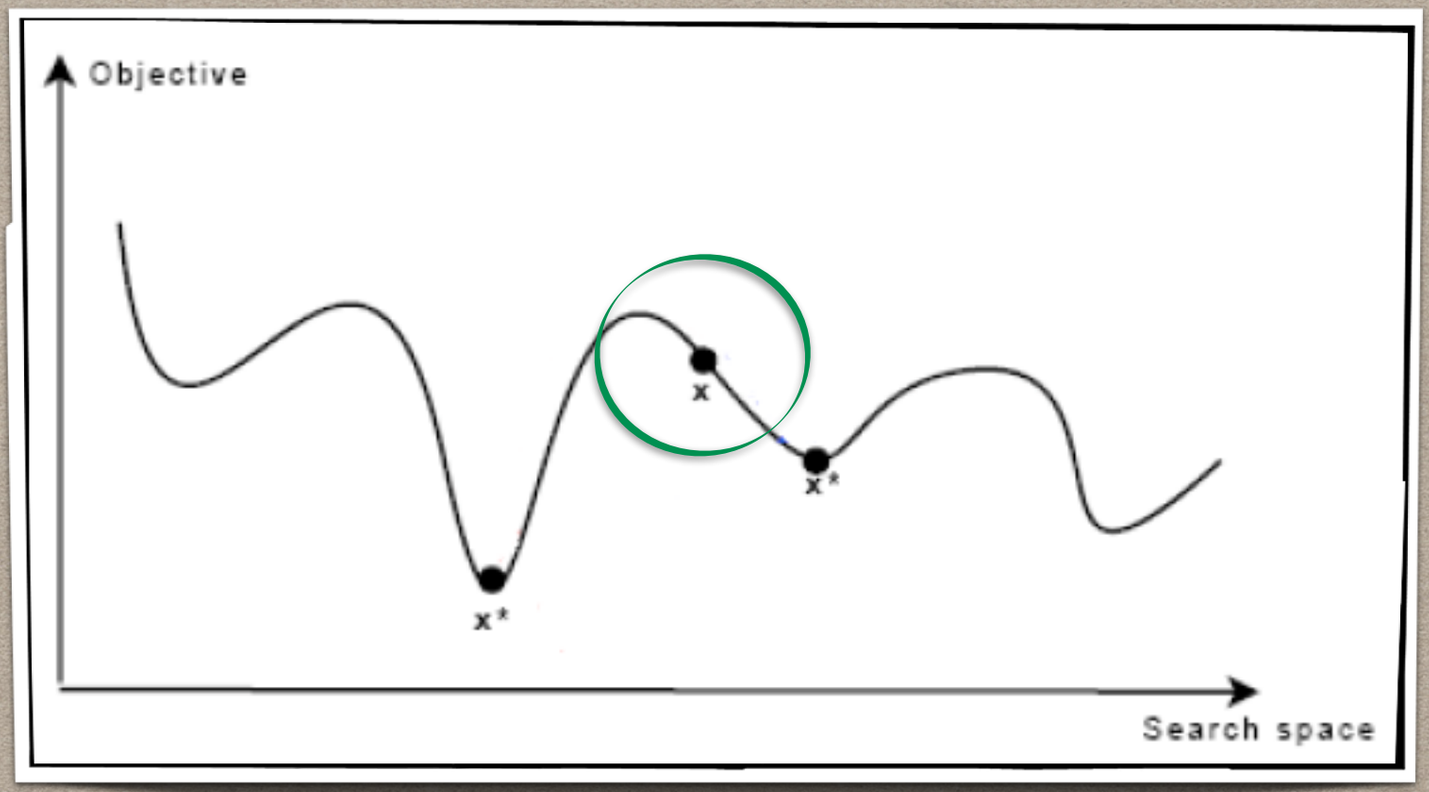


FIGURE 2.9 Inversion operator.

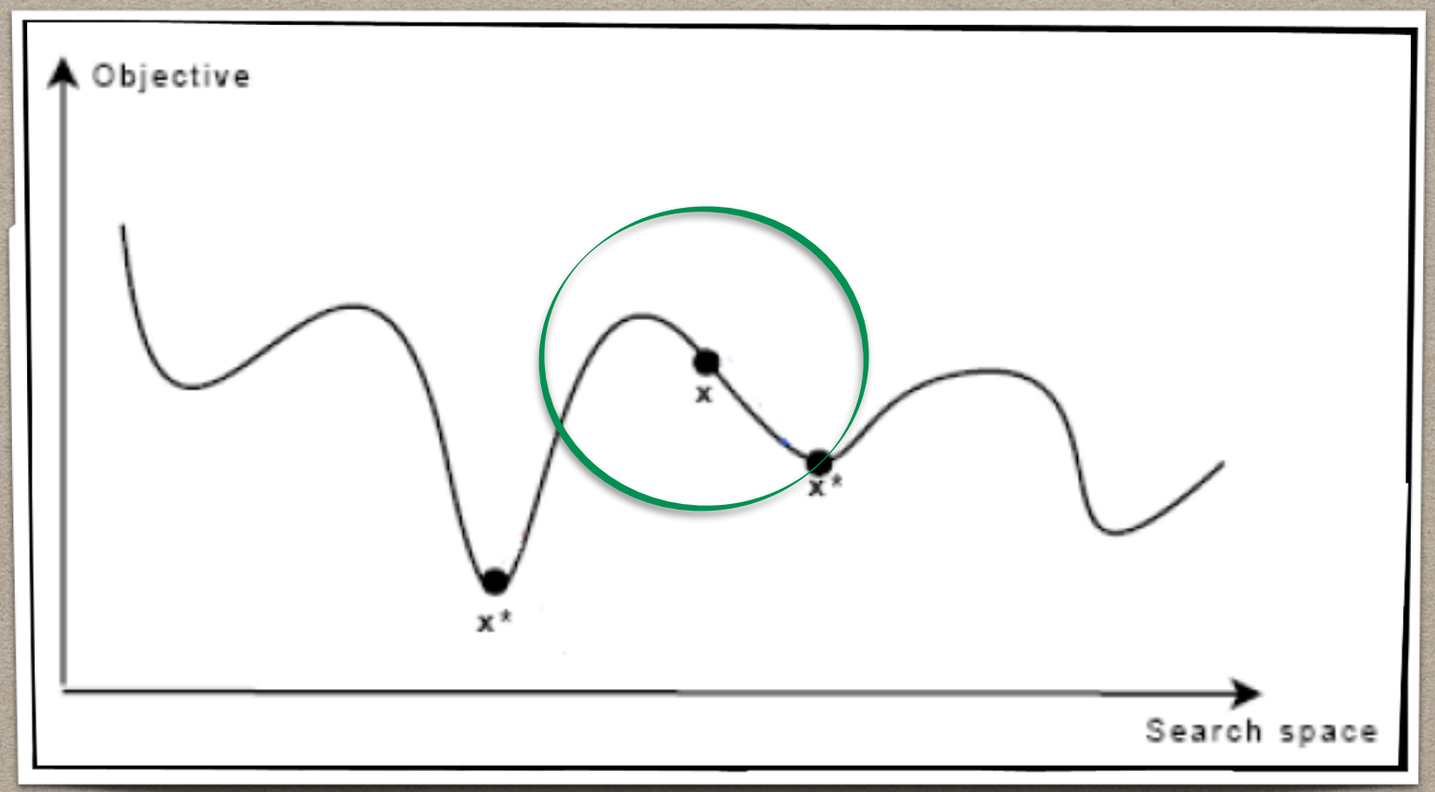
NEIGHBORHOOD



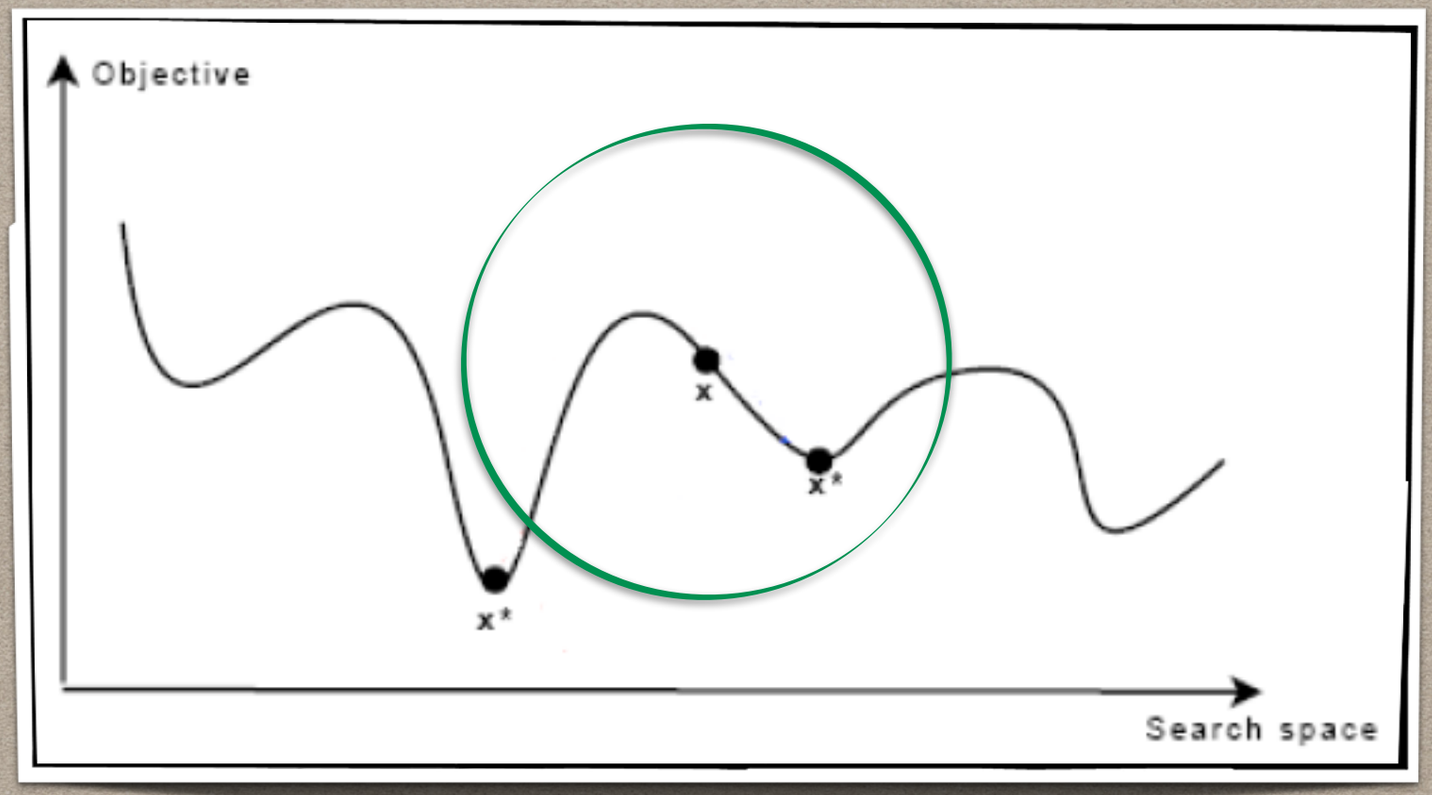
NEIGHBORHOOD



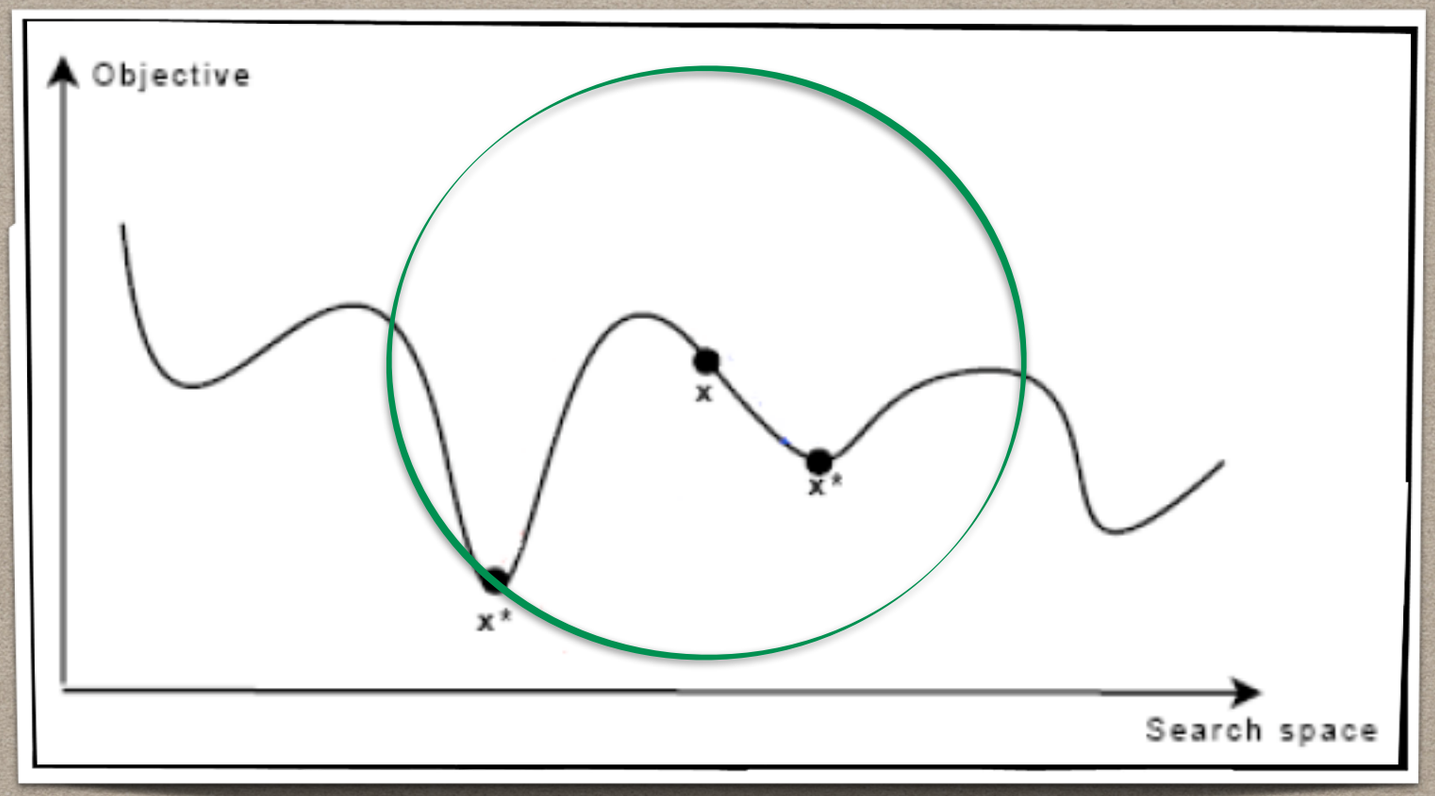
NEIGHBORHOOD



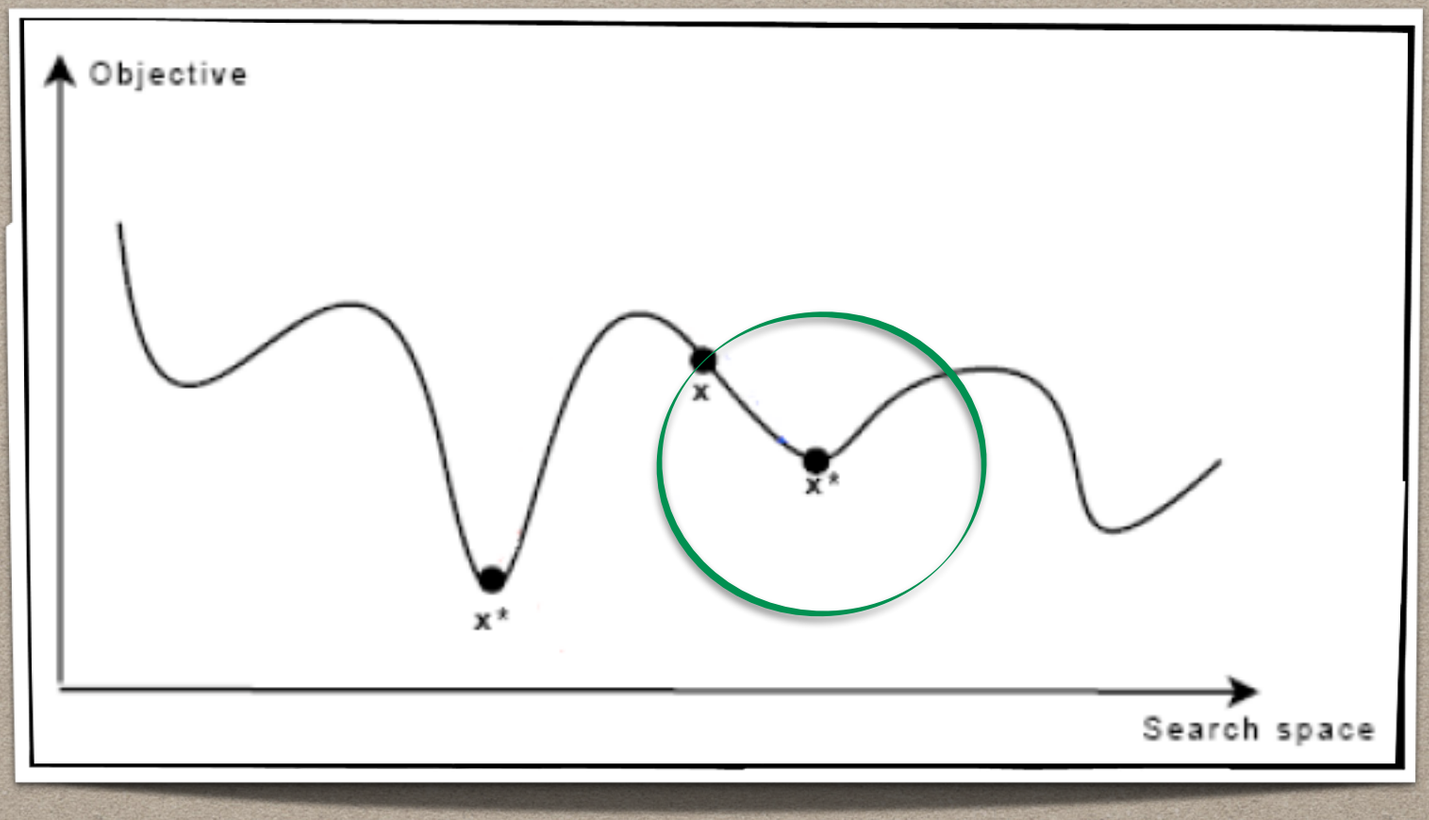
NEIGHBORHOOD



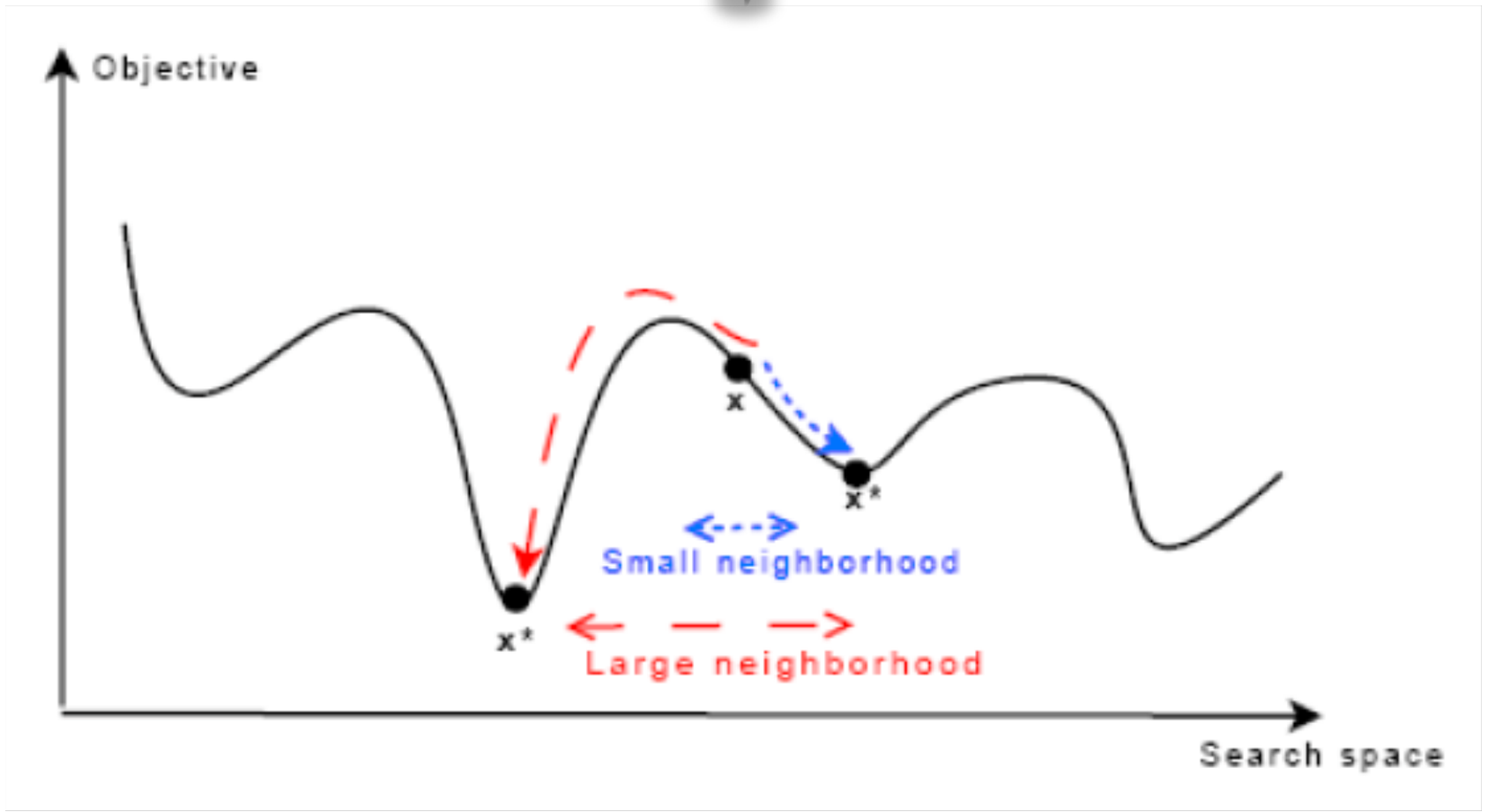
NEIGHBORHOOD



NEIGHBORHOOD



NEIGHBORHOOD



VERY LARGE NEIGHBORHOOD

Compromise: between size & quality of the neighborhood and computational complexity

Designing large neighborhood may improve the quality of the obtained solutions

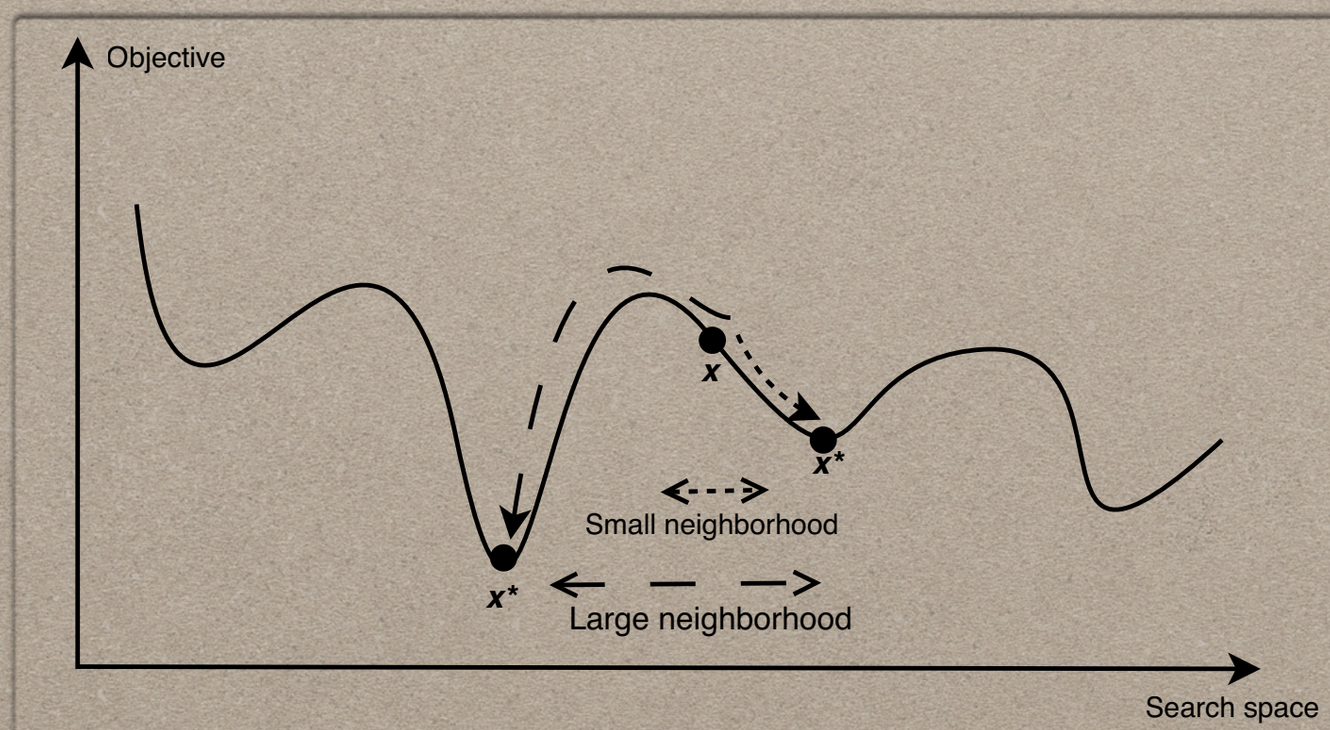


FIGURE 2.10 Impact of the size of the neighborhood in local search. Large neighborhoods improving the quality of the search with an expense of a higher computational time.

Disadvantage: additional computational time!!

The complexity of the search will be much higher.

Design efficient procedures to explore large neighborhoods

The size of the neighborhood for a solution s is the number of neighboring solutions of s