RESEARCH GROUP

Heuristics & Metaheuristics for Optimization & Learning

Mario Pavone

mpavone@dmi.unict.it

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

AA. 2024/2025 - LM 18

TABU SEARCH

Algorithm 2.9 Template of tabu search algorithm.

$s = s_0$; /* Initial solution */
Initialize the tabu list, medium-term and long-term memories ;
Repeat
  Find best admissible neighbor s' ; /* non tabu or aspiration criterion holds */
  $s = s'$;
  Update tabu list, aspiration conditions, medium and long term memories ;
  If intensification_criterion holds **Then** intensification ;
  If diversification_criterion holds **Then** diversification ;
Until Stopping criteria satisfied
Output: Best solution found.

ITERATIVE METHODS

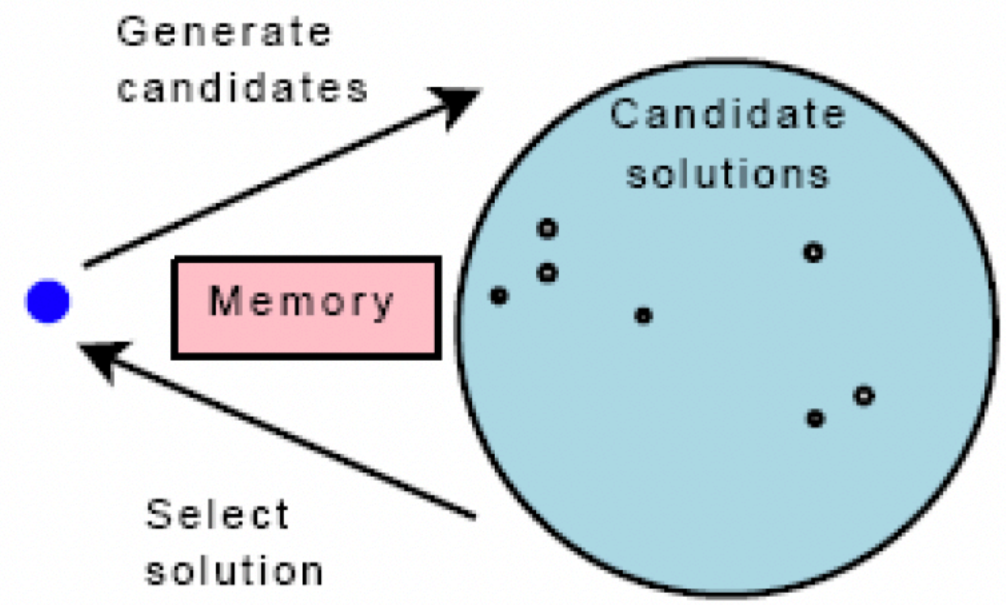
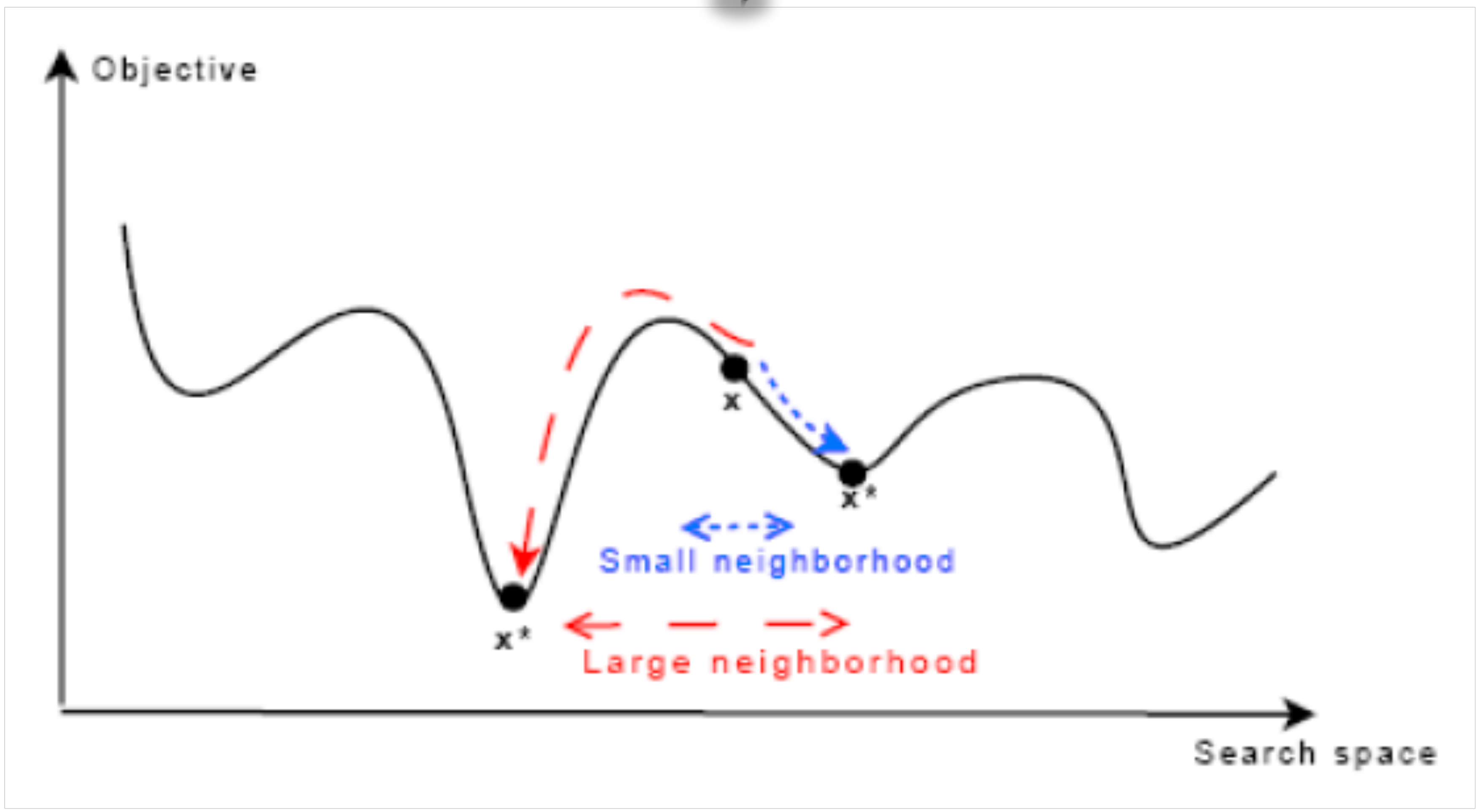


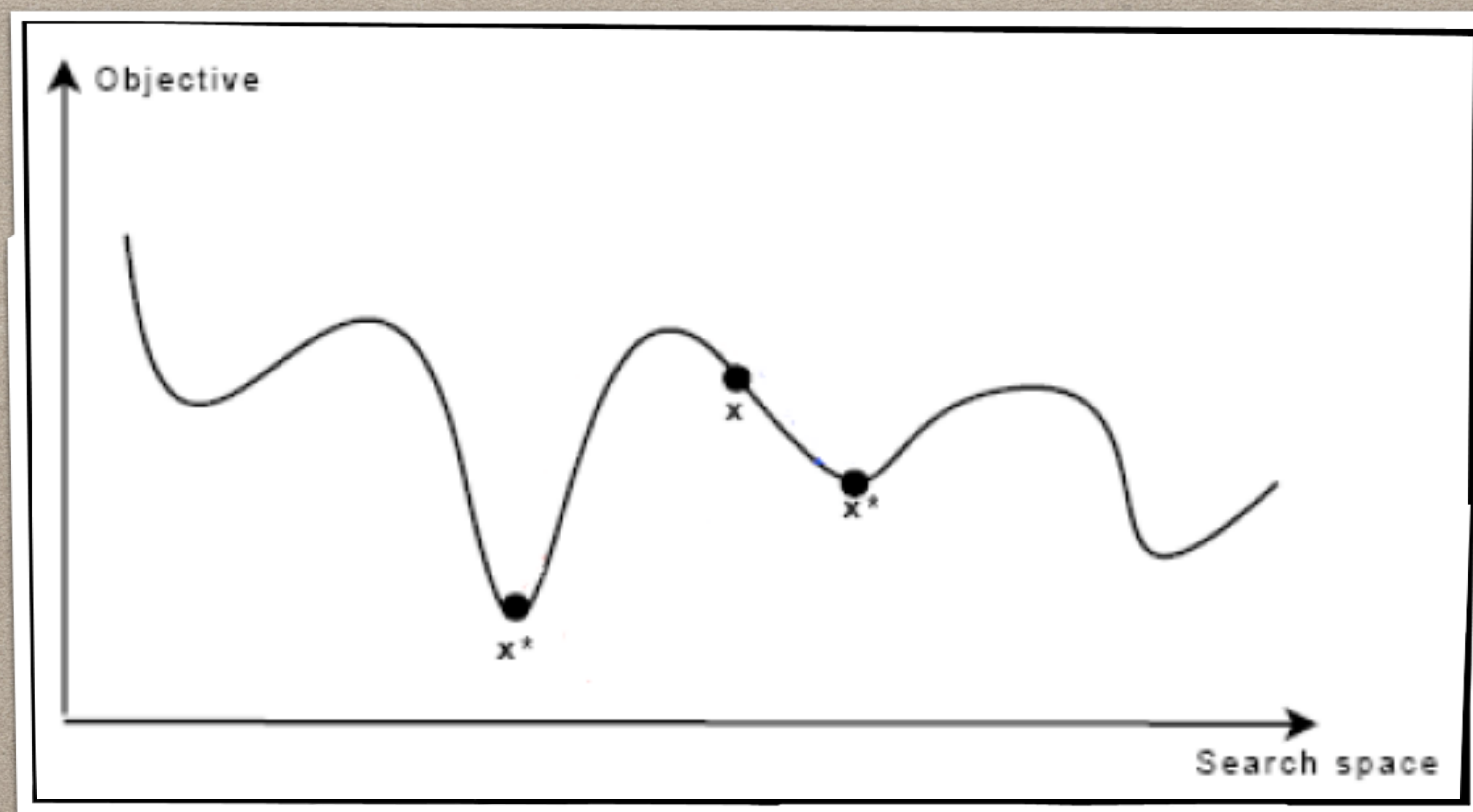
Fig. 2.1 Main principles of single-based metaheuristics.

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

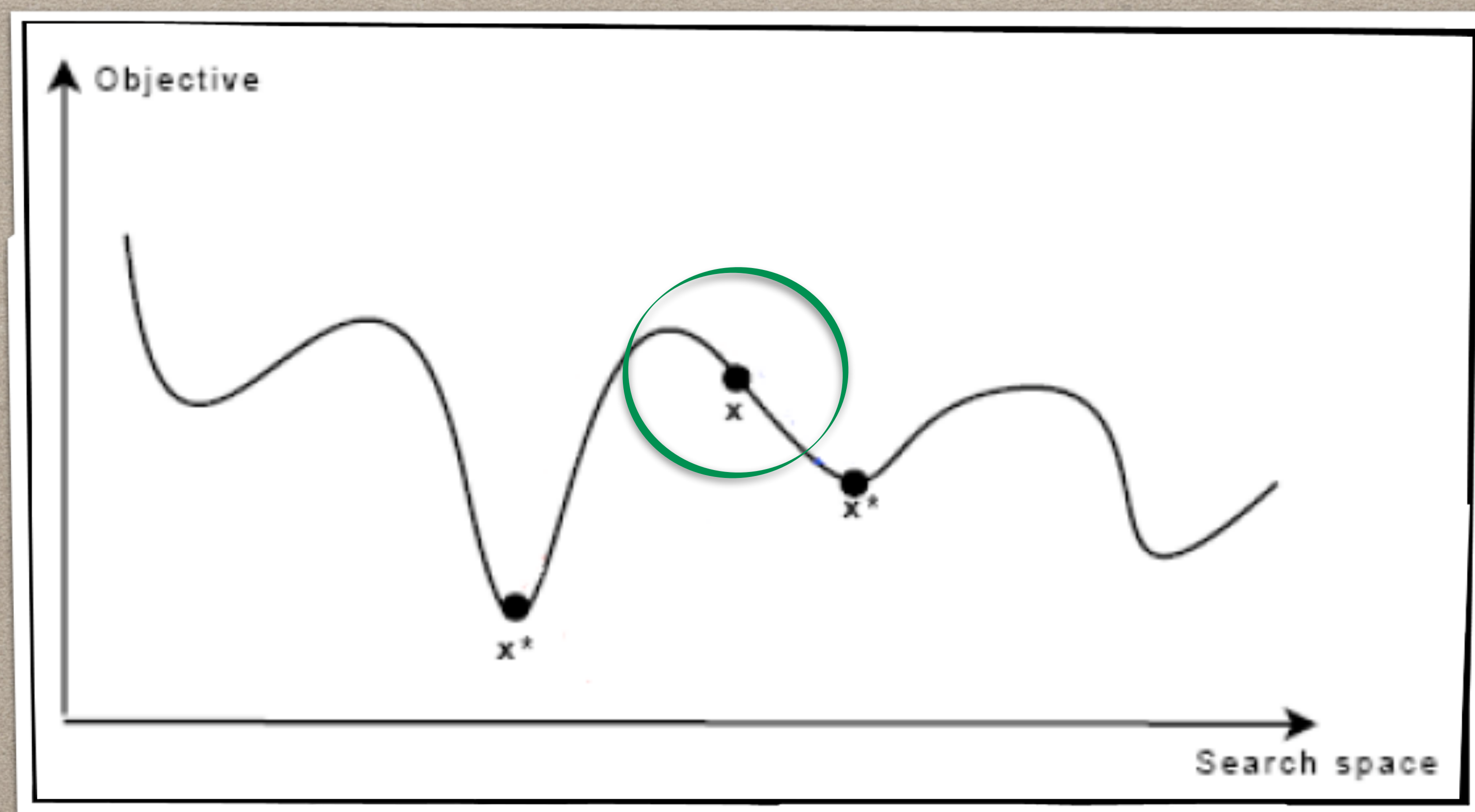
NEIGHBORHOOD SIZE IS CRUCIAL



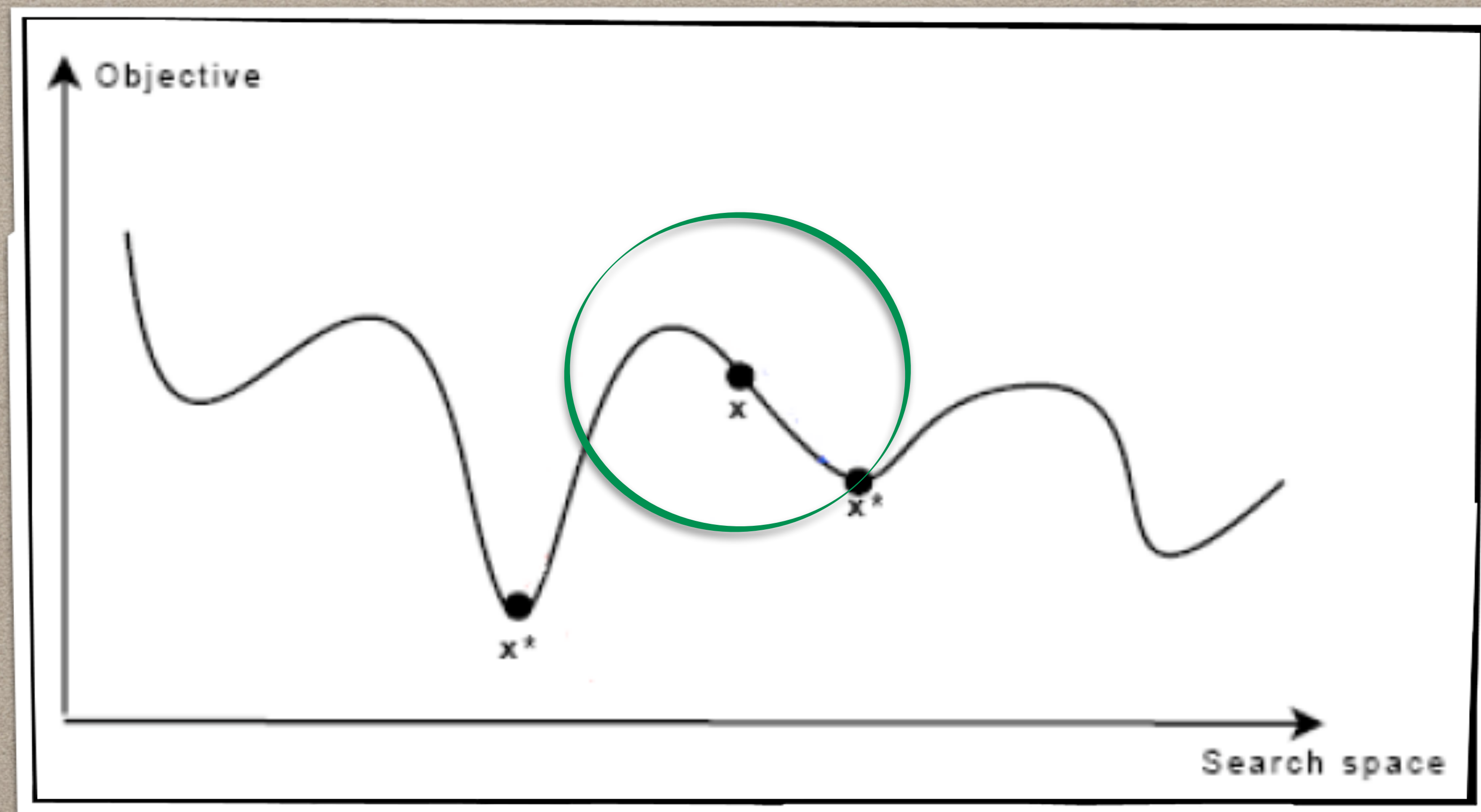
NEIGHBORHOOD SIZE IS CRUCIAL



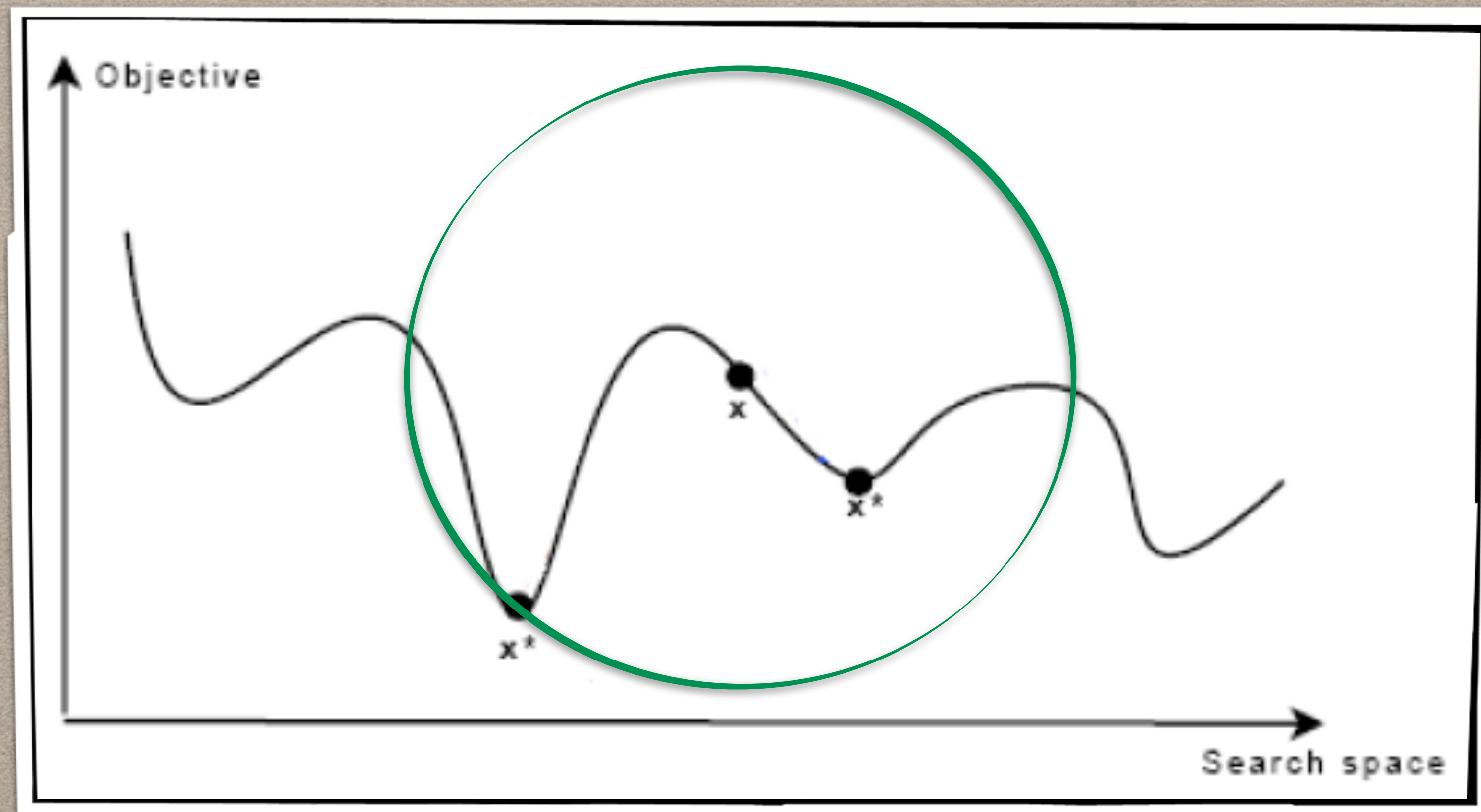
NEIGHBORHOOD SIZE IS CRUCIAL



NEIGHBORHOOD SIZE IS CRUCIAL



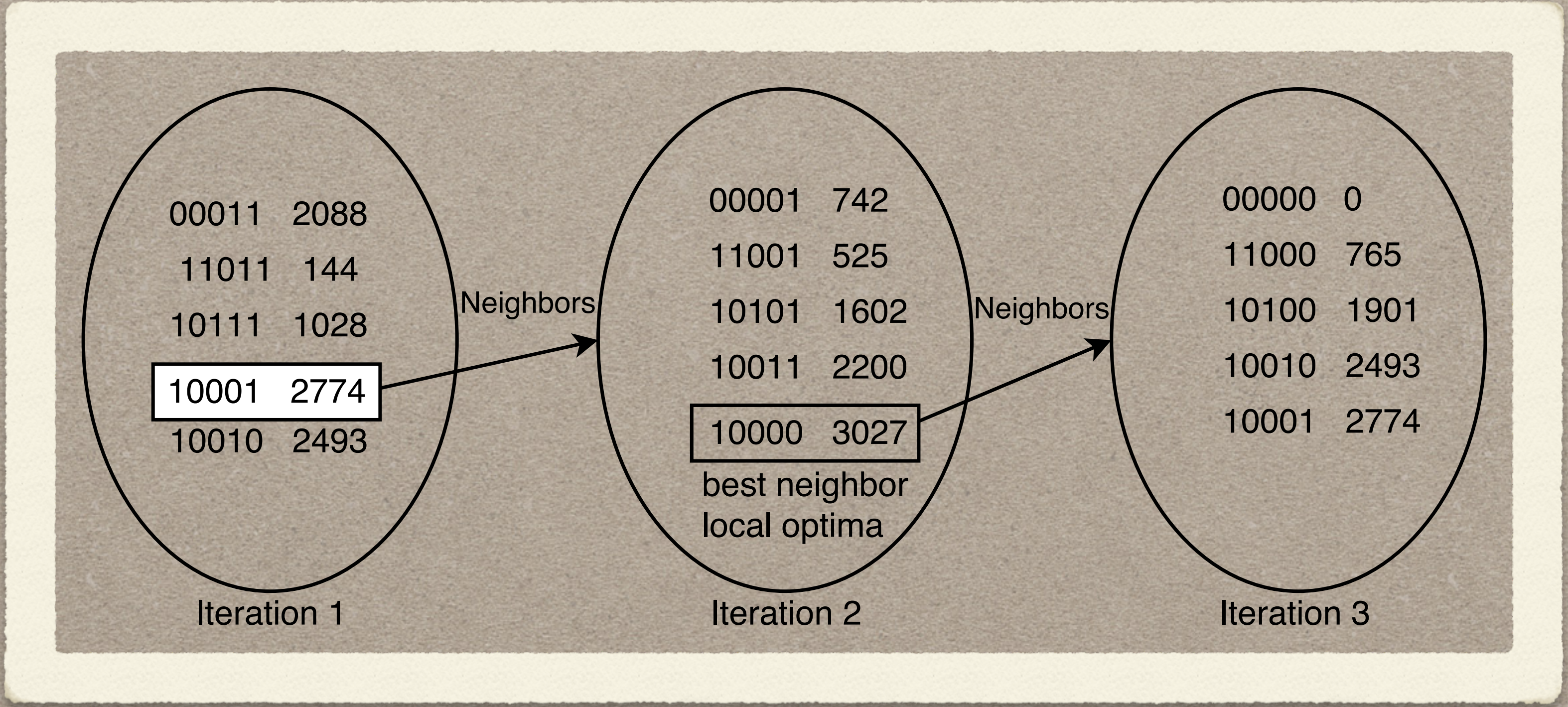
NEIGHBORHOOD SIZE IS CRUCIAL



LOCAL SEARCH

oldest and simplest metaheuristic method

Starts at a given initial solution, and at each iteration replaces the current solution by a neighbor that improves the objective function



The search stops when all candidate neighbors are worse than the current solution

LOCAL SEARCH

Algorithm 2.2 Template of a local search algorithm.

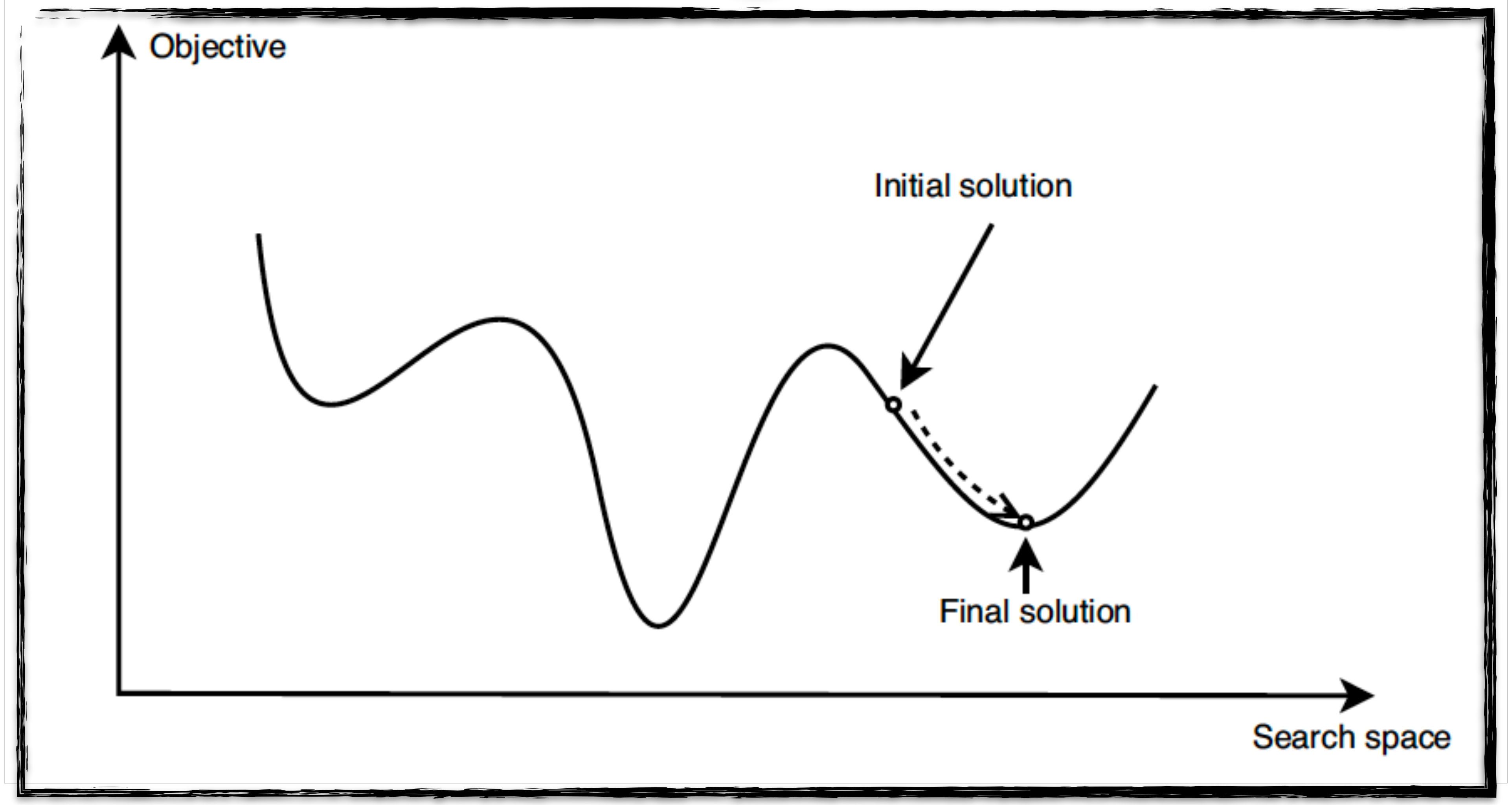
$s = s_0$; /* Generate an initial solution s_0 */
While not Termination_Criterion **Do**
  Generate $(N(s))$; /* Generation of candidate neighbors */
  If there is no better neighbor **Then** Stop ;
  $s = s'$; /* Select a better neighbor $s' \in N(s)$ */
Endwhile
Output Final solution found (local optima).

From an initial solution s_0, the algorithm will generate a sequence $s_1, s_2, \ldots, s_k$ of solutions, such that:

- The size of the sequence k is unknown a *priori*.
- $s_{i+1} \in N(s_i), \forall i \in [0, k-1]$.

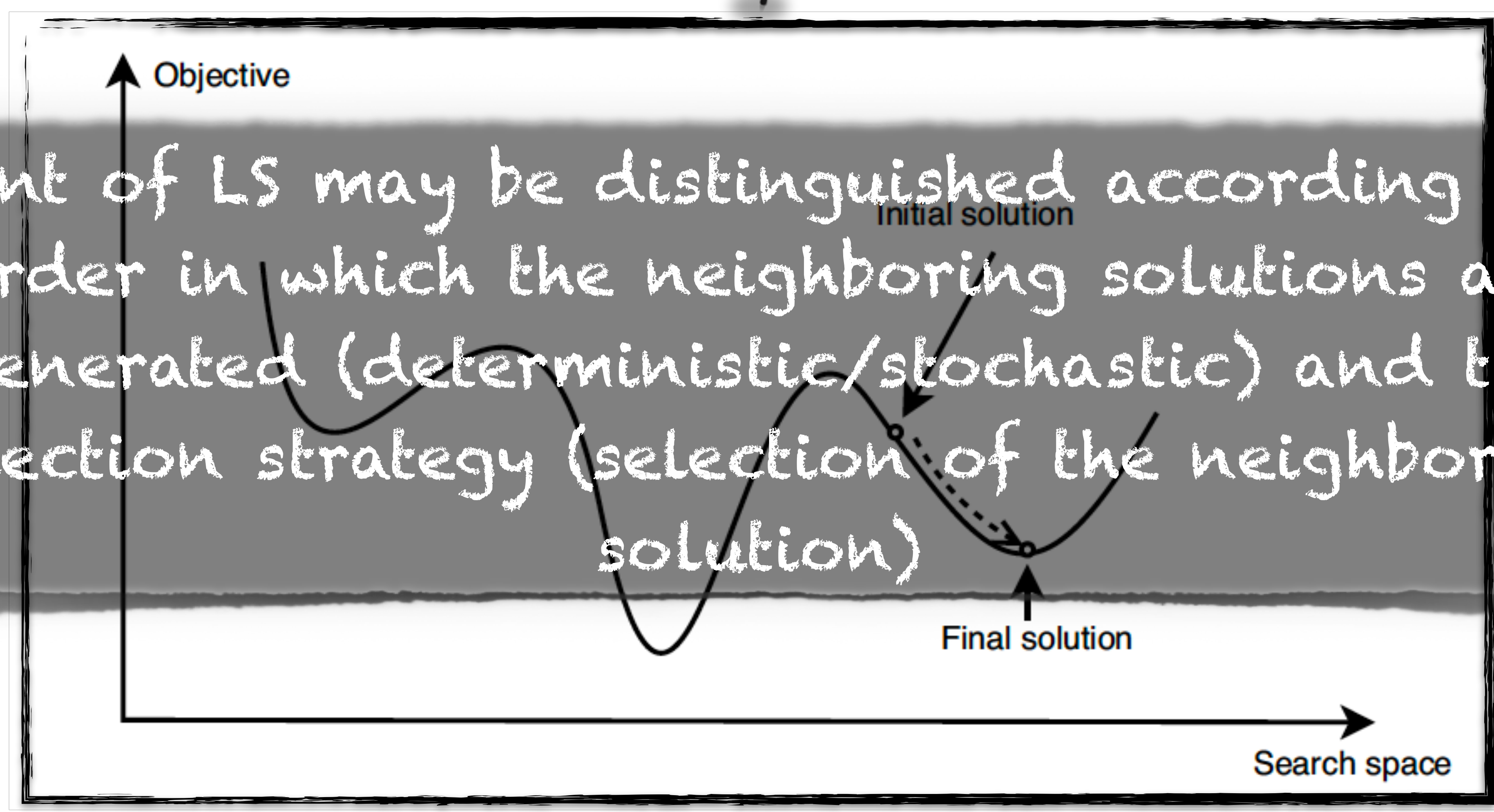
- $f(s_{i+1}) < f(s_i), \forall i \in [0, k-1]$.[13]
- s_k is a local optimum: $f(s_k) \leq f(s), \forall s \in N(s_k)$.

LOCAL OPTIMA VS GLOBAL OPTIMA



The quality of the local optima obtained by a Local Search method depends on the Initial Solution

LOCAL SEARCH



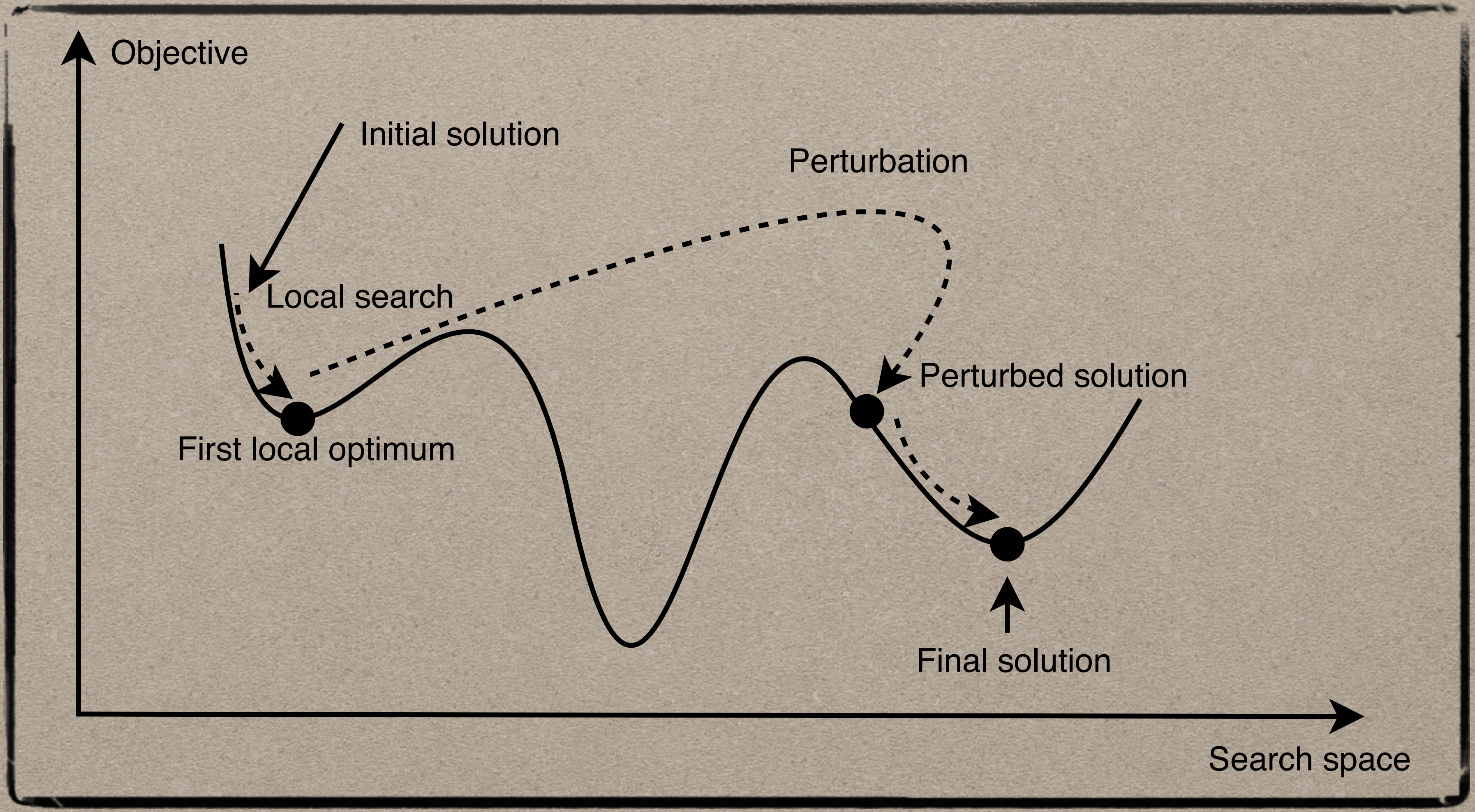
Variant of LS may be distinguished according to the order in which the neighboring solutions are generated (deterministic/stochastic) and the selection strategy (selection of the neighboring solution)

SELECTION OF NEIGHBOR

- **Best improvement (steepest descent):** In this strategy, the best neighbor (i.e., neighbor that improves the most the cost function) is selected. The neighborhood is evaluated in a fully deterministic manner. Hence, the exploration of the neighborhood is *exhaustive*, and all possible moves are tried for a solution to select the best neighboring solution. This type of exploration may be time-consuming for large neighborhoods.
- **First improvement:** This strategy consists in choosing the first improving neighbor that is better than the current solution. Then, an improving neighbor is immediately selected to replace the current solution. This strategy involves a partial evaluation of the neighborhood. In a *cyclic* exploration, the neighborhood is evaluated in a deterministic way following a given order of generating the neighbors. In the worst case (i.e., when no improvement is found), a complete evaluation of the neighborhood is performed.
- **Random selection:** In this strategy, a random selection is applied to those neighbors improving the current solution.

ITERATED LOCAL SEARCH

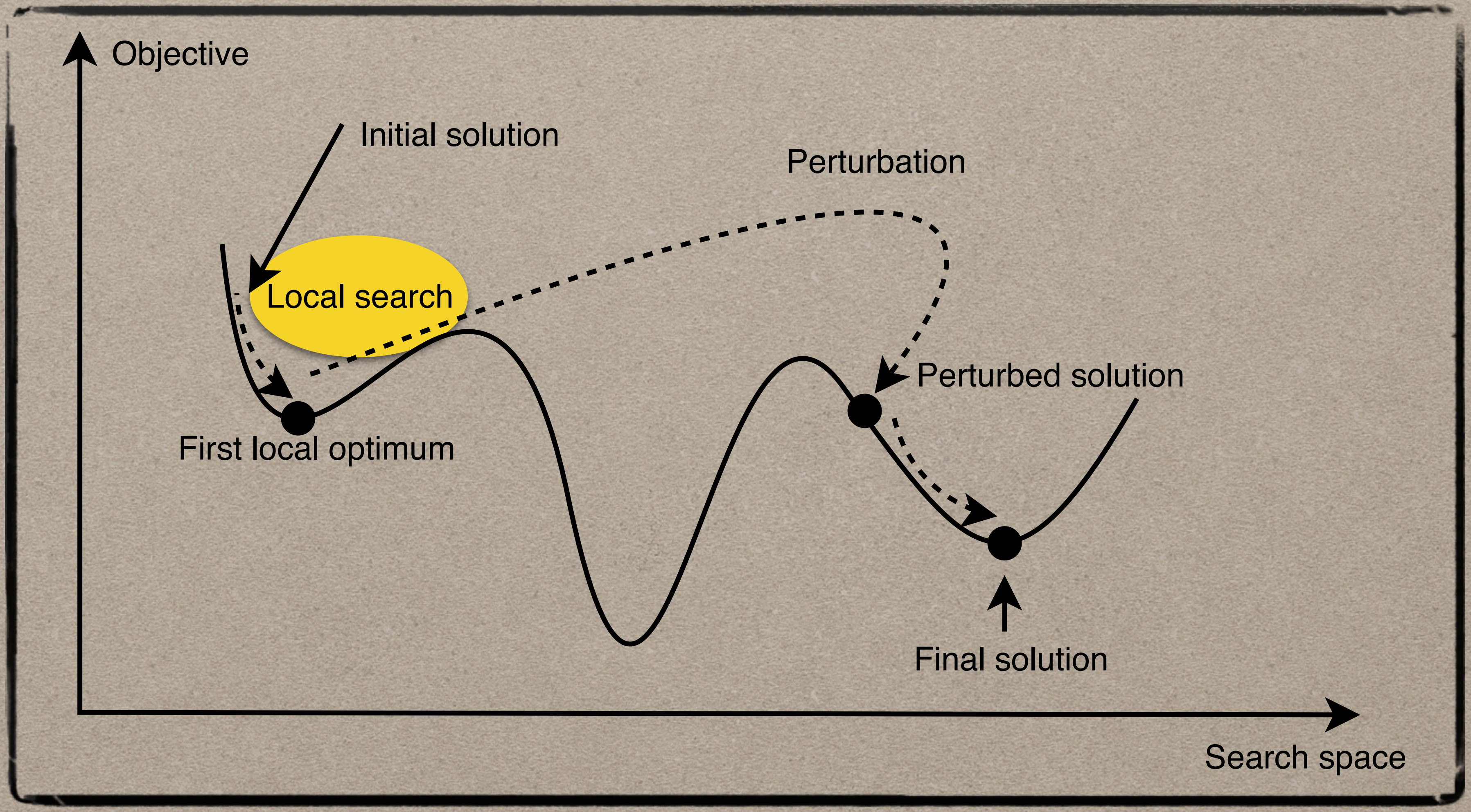
Three basic elements compose an ILS



ITERATED LOCAL SEARCH

Three basic elements compose an ILS

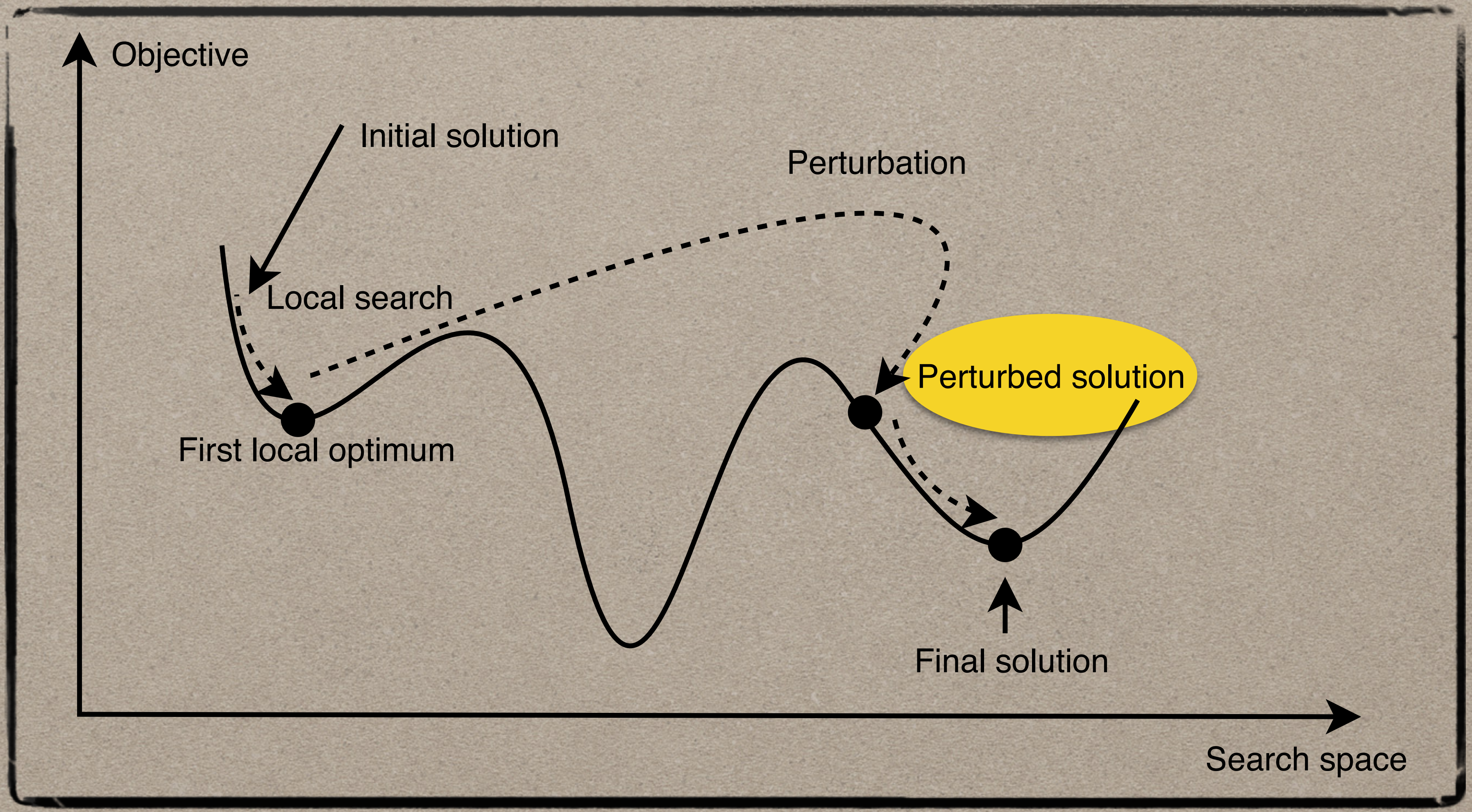
1) a Local Search is applied to an initial solution



ITERATED LOCAL SEARCH

Three basic elements compose an ILS

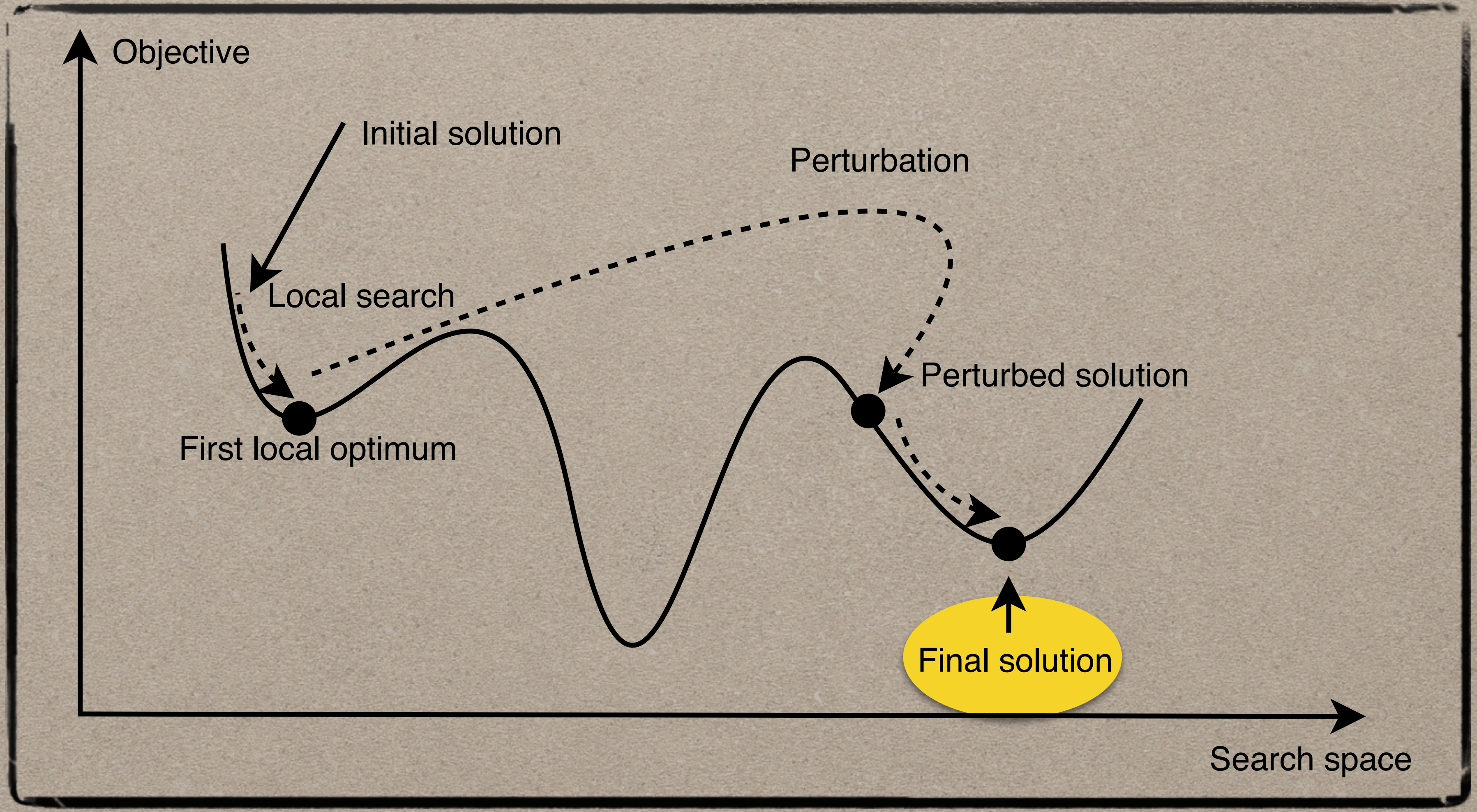
2) at each iteration a perturbation of the obtained local optima is carried out.



ITERATED LOCAL SEARCH

Three basic elements compose an ILS

3) a Local Search is applied to the perturbed solution



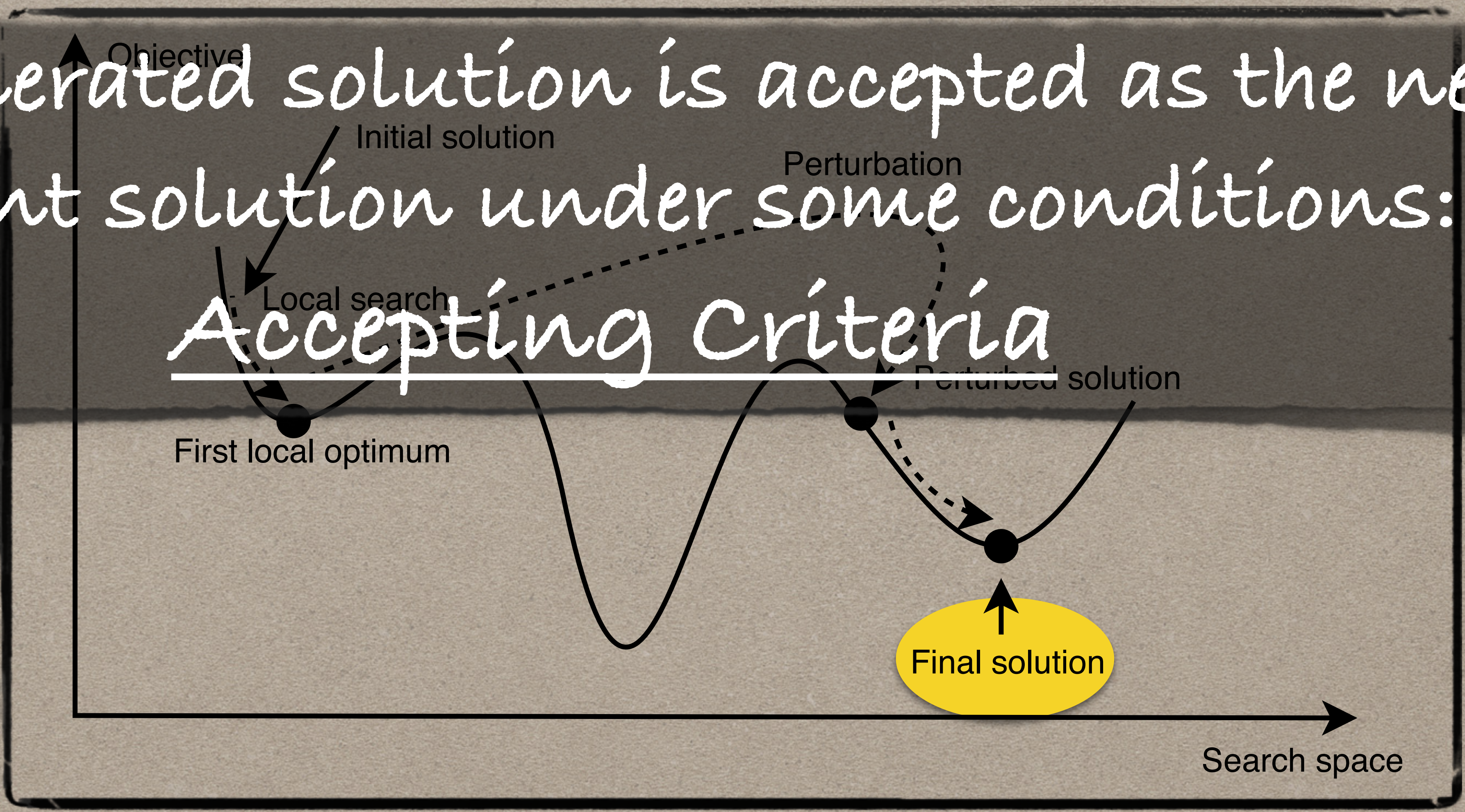
ITERATED LOCAL SEARCH

Three basic elements compose an ILS

3) a Local Search is applied to the perturbed solution

The generated solution is accepted as the new current solution under some conditions:

Accepting Criteria



ITERATED LOCAL SEARCH

Three basic elements compose an ILS

- **Local search:** Any S-metaheuristic (deterministic or stochastic) can be used in the ILS framework such as a simple descent algorithm, a tabu search, or simulated annealing. The search procedure is treated as a black box

- **Perturbation Method.** The perturbation operator may be seen as a large random move of the current solution. The perturbation method should keep some part of the solution and perturb strongly another part of the solution to move hopefully to another basin of attraction.

- **Acceptance criteria.** The acceptance criterion defines the conditions the new local optima must satisfy to replace the current solution.

ITERATED LOCAL SEARCH

Three basic elements compose an ILS

- **Local search:** Any S-metaheuristic (deterministic or stochastic) can be used in the ILS framework such as a simple descent algorithm, a tabu search, or simu-[illegible]

NOTE: the design of ILS will depend mainly on the used perturbation method and the acceptance criterion

- **Perturbation Method.** The perturbation operator may be seen as a large random move of the current solution. The perturbation method should keep some part of the solution and perturb strongly another part of the solution to move hopefully to another basin of attraction.

- **Acceptance criteria.** The acceptance criterion defines the conditions the new local optima must satisfy to replace the current solution.

ITERATED LOCAL SEARCH

Algorithm 2.10 Template of the iterated local search algorithm.

$s_* =$ local search(s_0) ; /* Apply a given local search algorithm */
Repeat
  $s' =$ Perturb (s_*, search history) ; /* Perturb the obtained local optima */
  $s'_* =$ Local search (s') ; /* Apply local search on the perturbed solution */
  $s_* =$ Accept (s_*, s'_*, search memory) ; /* Accepting criteria */
Until Stopping criteria
Output: Best solution found.

PERTURBATION METHOD

The length of a perturbation may be related to the neighborhood associated with the encoding or with the number of modified components

A too SMALL perturbation => may generate cycles in the search and no gain is obtained

Probability to explore other basins of attraction is low

Too LARGE perturbation => erase the information about the search memory and the good properties of the local optima are skipped

ACCEPTANCE CRITERIA

The role of the acceptance criterion combined with the perturbation method is to control the classical trade-off between the intensification and the diversification tasks. The first extreme solution in terms of intensification is to accept only improving solutions in terms of the objective function (strong selection). The extreme solution in terms of diversification is to accept any solution without any regard to its quality (weak selection). Many acceptance criteria that balance the two goals may be applied:

- **Probabilistic acceptance criteria:** Many different probabilistic acceptance criteria can be found in the literature. For instance, the Boltzmann distribution of simulated annealing. In this case, a cooling schedule must be defined.

- **Deterministic acceptance criteria:** Some deterministic acceptance criteria may be inspired from the threshold accepting algorithms and the related algorithms, such as the great deluge and the record-to-record algorithms.