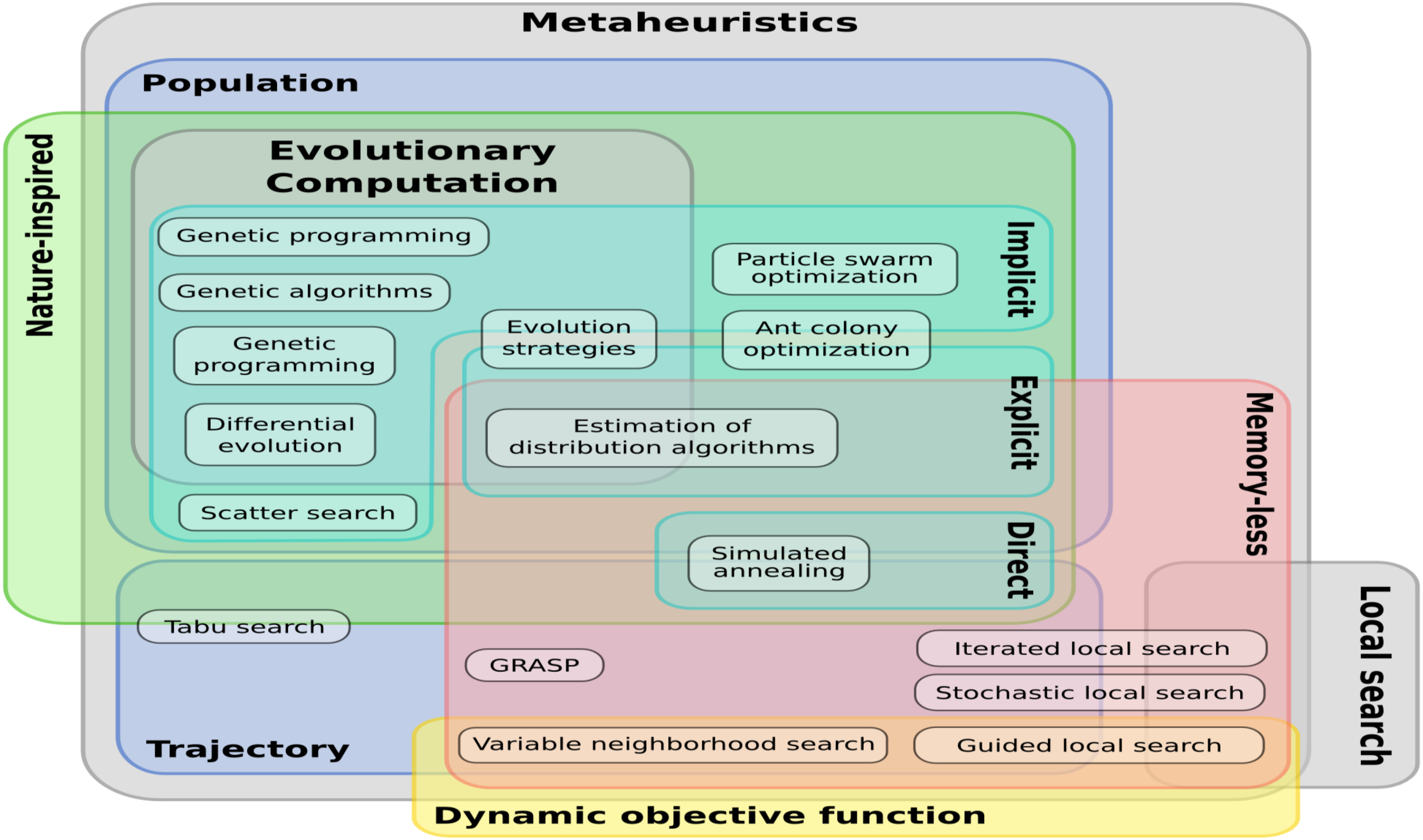
METAHEURISTICS

MAIN COMMON CONCEPTS FOR METAHEURISTICS:

- Classification
- Complexity
- Solution Representation
- Objective Function

Example of meta-heuristics Algorithm

Adopted from *Dero 2002*



Classification of Meta-heuristic Algorithms

Nature inspired
versus
nonnature inspired

Nature inspired

From Biology:
- Evolutionary algorithms
- Artificial immune systems

From Swarm Intelligence:
- Ants colonies
- bees colonies
- Particle swarm optimization

From Physics:
- Simulated annealing

Nonnature inspired

- Tabu search algorithms

Classification of Meta-heuristic Algorithms

Memory usage
versus
Memoryless

Memoryless Algorithms:
No information extracted dynamically is used during the search.

- Local search.
- Simulated annealing.

Memory Algorithms:
Some information extracted online is used during the search.

- Tabu search.

Classification of Meta-heuristic Algorithms

Deterministic versus Stochastic

Deterministic Algorithms:

Solves an optimization problem by making deterministic decisions .

- Local search.
- Tabu search.

Stochastic Algorithms:

Some random rules are applied during the search.

- Simulated annealing.
- Evolutionary algorithms.

Note:

In deterministic algorithms, using the same initial solution will lead to the same final solution, whereas in stochastic metaheuristics, different final solutions may be obtained from the same initial solution.

Classification of Meta-heuristic Algorithms

Iterative versus Greedy

<u>Iterative Algorithms</u>:
start with a complete solution (or population of solutions) and transform it at each iteration using some search operators.

<u>Greedy Algorithms:</u>
start from an empty solution, and at each step a decision variable of the problem is assigned until a complete solution is obtained.

<u>Note</u>:
Most of the metaheuristics are iterative algorithms.

Classification of Meta-heuristic Algorithms

Single solution based search
versus
Population-based search

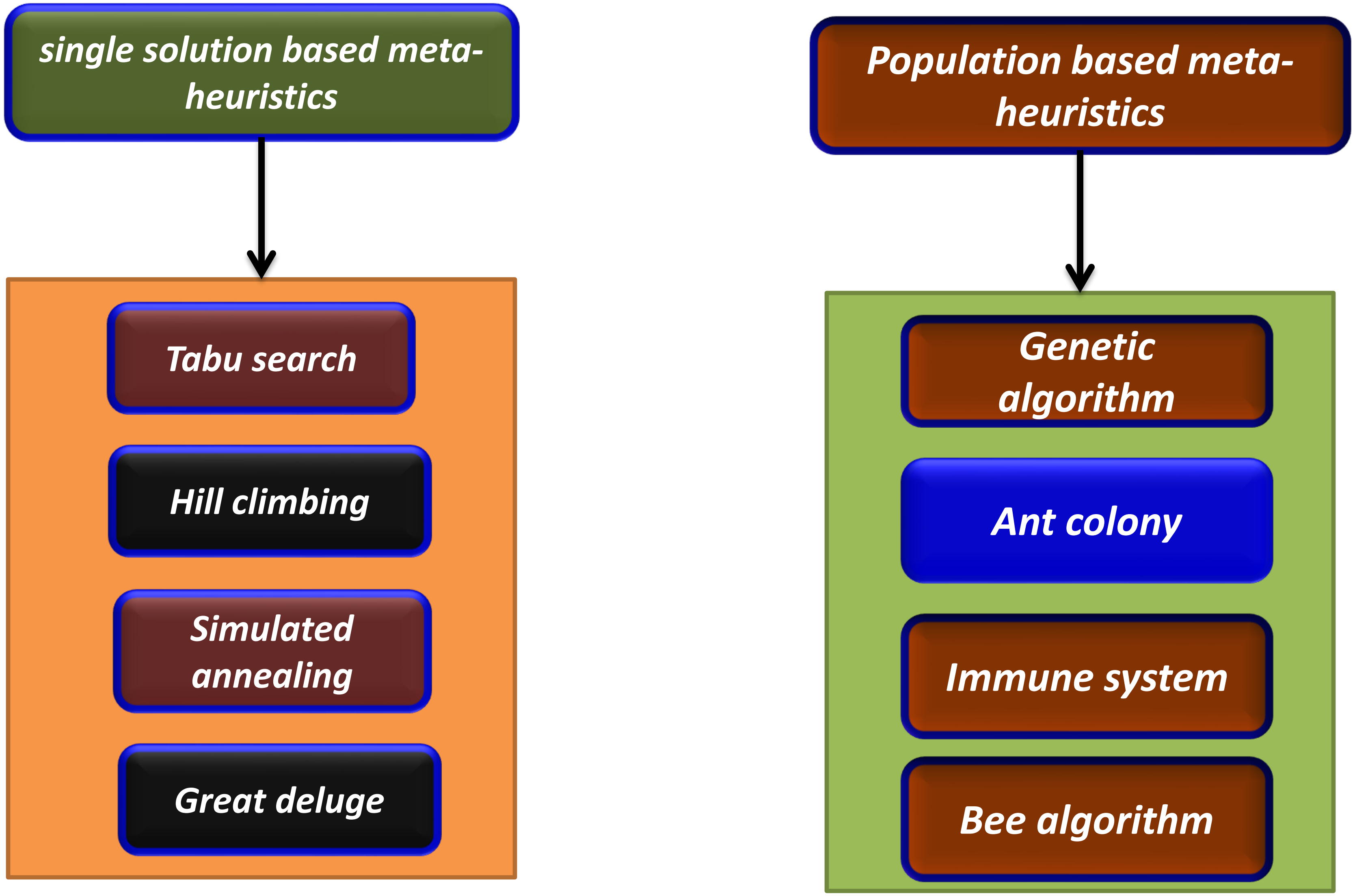
Single solution based search Algorithms:
Manipulate and transform a single solution during the search .
- Local search.
- Simulated annealing.

Population-based search Algorithms:
A whole population of solutions is evolved.
- Particle swarm.
- Evolutionary algorithms.

These two families have complementary characteristics:

- Single-solution based metaheuristics are **exploitation oriented**; they have the power to intensify the search in local regions.
- Population-based metaheuristics are **exploration oriented**; they allow a better diversification in the whole search space.

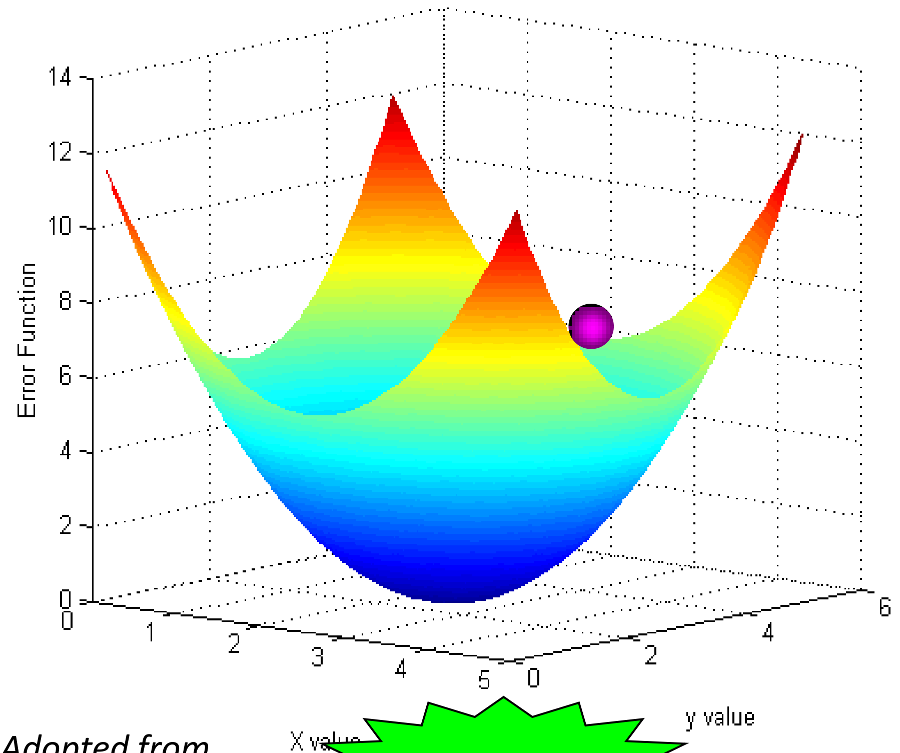
Meta-heuristic Algorithms

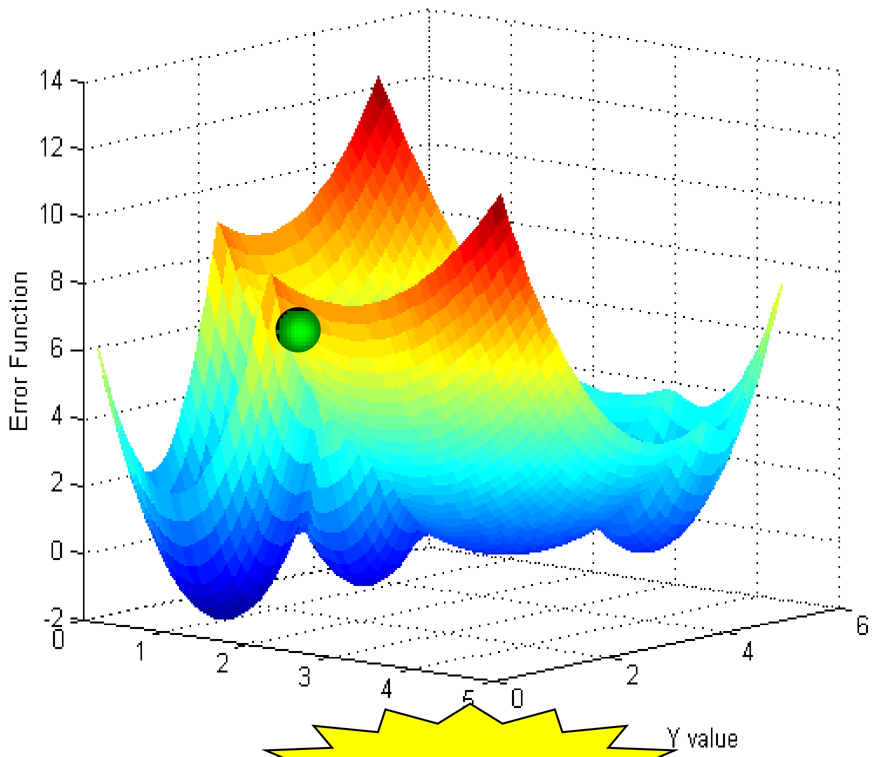


Single solution based meta-heuristics (local search algorithms)

Single nodal case

Multiple nodal case





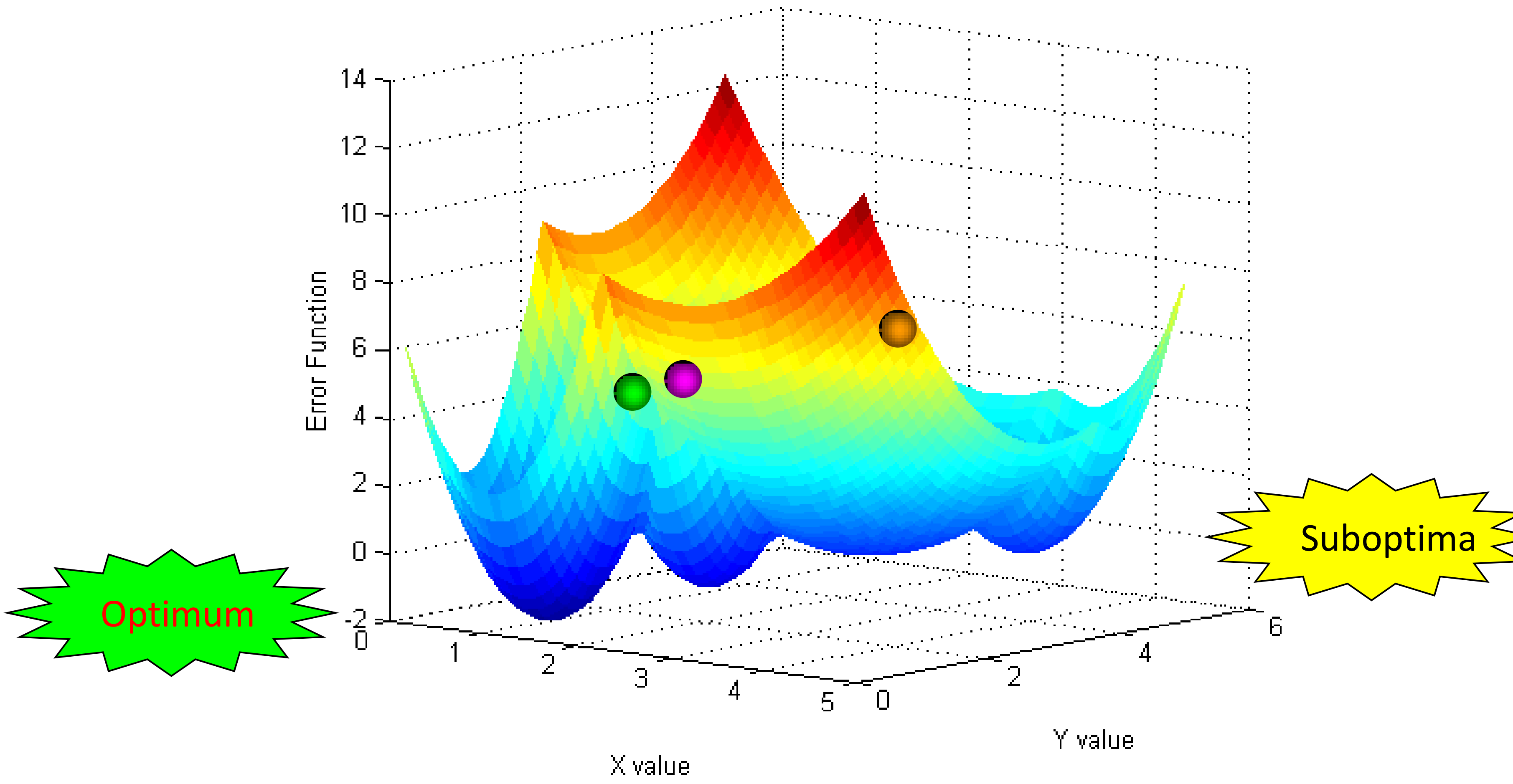
Optimum

Suboptimum

Adopted from Gwangju Institute Of Science and Technology (GIST)

Population based meta-heuristics

Multiple search or multiple solutions



Meta-heuristic Algorithms

Search Space

Is the set or domain through which an **algorithm searches and it** represents the set of feasible solutions among which the an optimal solution resides.

- Each point in the search space represents one possible solution.
- Each possible solution can be "marked" by its value (or fitness) for the problem.
- The problem is that the search can be very complicated. One may not know where to look for a solution or where to start.

Global optimum

A solution $s* \in S$ is a global optimum if it has a better objective function4 than all solutions of the search space, that is, $\forall s \in S, f(s*) \leq f(s)$.

Local optimum

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 $s' \in N(s)$.

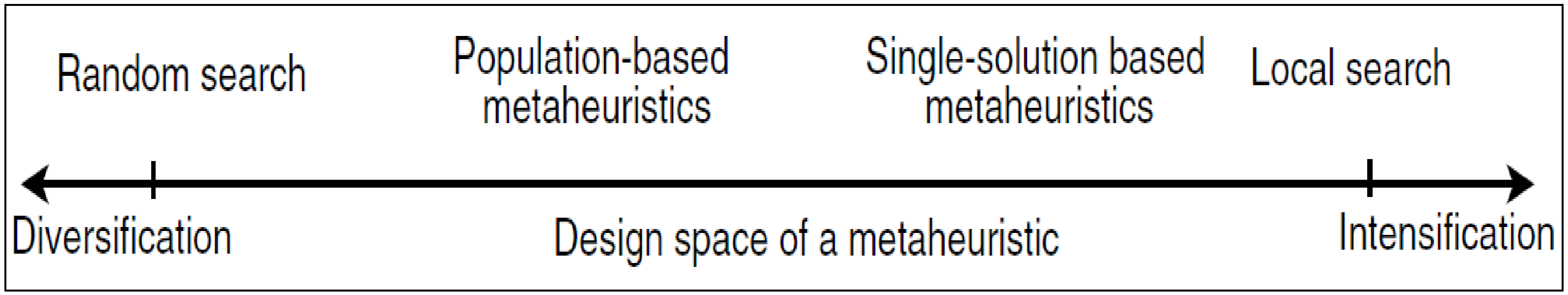
Diversification and Intensification

In designing a metaheuristic, two contradictory criteria must be taken into account:

exploration of the search space (diversification) and **exploitation** of the best solutions found (intensification)

In intensification, the promising regions are explored more thoroughly in the hope to find better solutions.

In diversification, nonexplored regions must be visited to be sure that all regions of the search space are evenly explored.


Random search
Population-based metaheuristics
Single-solution based metaheuristics
Local search
Diversification
Design space of a metaheuristic
Intensification

MAIN COMMON CONCEPTS FOR METAHEURISTICS

Solution Representation

Solution Representation plays a major role in the efficiency and effectiveness of any metaheuristic and constitutes an essential step in designing a metaheuristic.

The representation must be suitable and relevant to the tackled optimization problem

the efficiency of a representation is also related to the search operators applied on this representation (neighborhood, recombination, etc.)

The representation also determines how the objective function will be evaluated

MAIN COMMON CONCEPTS FOR METAHEURISTICS

The characteristics of Solution Representation

Completeness/Inclusion

- All solutions associated with the problem must be represented.
- whether or not the resulting solution space includes all the feasible solutions.
- If NOT (intentionally or unintentionally), may result in excluding some high quality solutions.

Connectivity

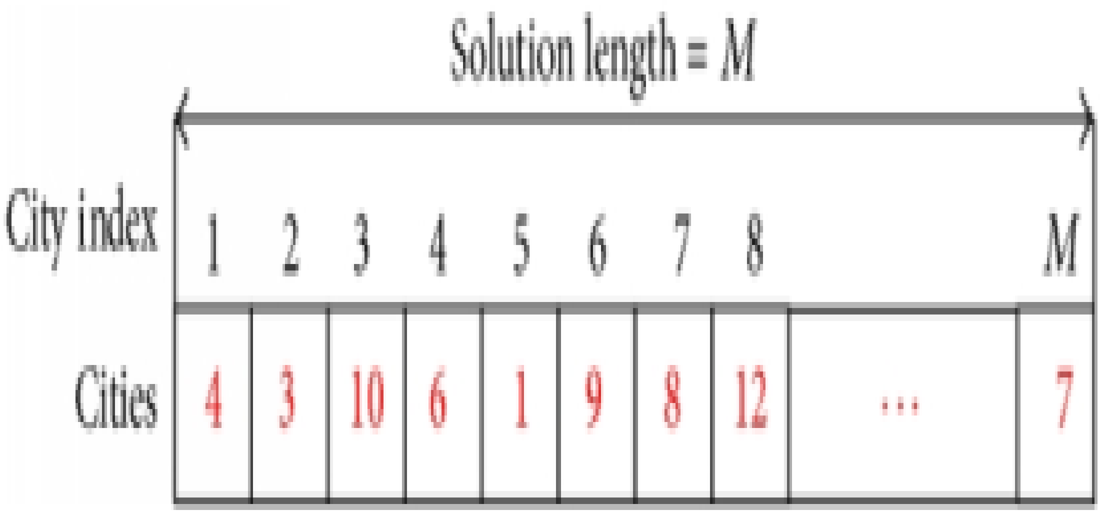
- A search path must exist between any two solutions of the search space. Any solution of the search space, especially the global optimum solution, can be attained.
- Any solution can be transformed into any other one by changing one value at a time

Efficiency

- The representation must be easy to manipulate by the search operators (evaluation, neighborhood, recombination, etc.).
- The time and space complexities of the operators dealing with the representation must be reduced.

MAIN COMMON CONCEPTS FOR METAHEURISTICS

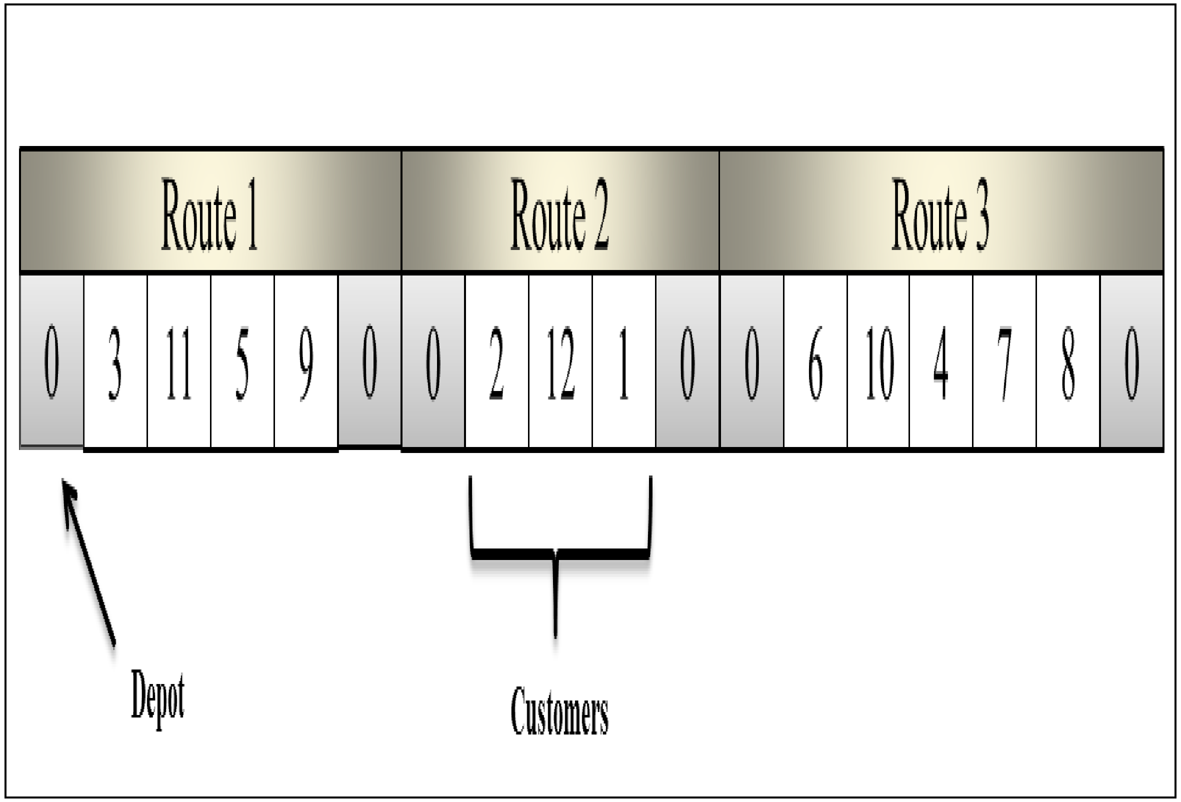
Solution Representation-Examples:



TSP solution representation

MAIN COMMON CONCEPTS FOR METAHEURISTICS

Solution Representation-Examples:



VRP solution representation

Solution Representation-Examples:

0	1	2	3	4	5	6	Item Number
0	1	0	1	1	0	1	Chromosome
2	9	8	5	4	0	2	Profit Values
7	5	3	1	5	9	8	Weight Values

Knapsack capacity = 15
Total associated profit = 18
Last item not picked as it exceeds knapsack capacity

Knapsack solution representation

Solution Representation-Examples:

$$f(x) = 2x + 4x \cdot y - 2x \cdot z$$

1.23	5.65	9.45	4.76	8.96

Vector of real values

Continuous optimization solution representation

MAIN COMMON CONCEPTS FOR METAHEURISTICS

Objective Function

The objective function is an important element in designing a metaheuristic as It will guide the search toward “good” solutions of the search space

- formulates the goal to achieve.
- Associates with each solution a real value to describe its quality or fitness.
- If the objective function is improperly defined, it can lead to non-acceptable solutions whatever metaheuristic is used.

Thank U

MAIN COMMON CONCEPTS FOR METAHEURISTICS

Complexity

Any algorithm needs two important resources to solve a problem: time and space.

The effort of an optimization method can be measured as the **time** (computation time) and **space** (computer memory) that is consumed by the method.

Time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input.

Space complexity of an algorithm quantifies the amount of space or memory taken by an algorithm to run as a function of the length of the input.

MAIN COMMON CONCEPTS FOR METAHEURISTICS

Complexity

complexity is concerned about how a particular algorithm performs.
without depending on the implementation details (factors);
HARDWARE, OPERATING SYSTEM, PROCESSORS, etc

- estimate efficiency of each algorithm asymptotically.
- **Time complexity** T(n) measure as the number of "steps“- such step takes constant time ***c***

MAIN COMMON CONCEPTS FOR METAHEURISTICS

Complexity

Note: To compute complexity we will ignore the lower order terms.

for example:

Let f(N)=2*N^2+3*N+5
O(f(N))=O(2*N^2+3*N+5)=O(N^2)

Complexity – Examples:

- **Constant Time: O(1)**

An algorithm is said to run in constant time if it requires the same amount of time regardless of the input size; accessing any element in array.

- **Linear Time: O(n)**

An algorithm is said to run in linear time if its time execution is directly proportional to the input size; find if an integer exists in a given array?

MAIN COMMON CONCEPTS FOR METAHEURISTICS

Complexity – Examples:

- **Quadratic Time: O(n^2):**
  An algorithm is said to run in Quadratic time if its time execution is proportional to the square of the input size;
  bubble sort and selection sort algorithms.

- **Logarithmic Time: O(log n):**
  An algorithm is said to run in logarithmic time if its time execution is proportional to the logarithm of the input size;
  Algorithms does not use the whole input- Binary search algorithm

MAIN COMMON CONCEPTS FOR METAHEURISTICS

Complexity – Examples:

for (int i = 0; i < N; i++)	$O(N)$

for (int i = 0; i < N; i++)
for (int j = 0; j < N; j++)
	$O(N^2)$

for (int i = 0; i < N; i++)
for (int j = 0; j < i; j++)
	$O(N^2)$

for (int i = N; i >0; i=i/2)
for (int j = 0; j < i; j++)
	$O(N)$