1. Introduction

Fall 2010

Instructor: Dr. Masoud Yaghini

Optimization Problems

Optimization problems

- The many decision-making problems can be often expressed as an constrained **optimization problem** with some decision variables that are restricted by a set of constraints.
- Types of constrained optimization problems:
  - **Combinatorial problems**: When the decision variables are discrete
  - **Continuous problems**: When the decision variables are continuous
  - **Mixed problems**

Combinatorial Problems

- Examples of real-world combinatorial optimization problems include:
  - Assembly-line balancing problems
  - Vehicle routing and scheduling problems
  - Facility location problems
  - Facility layout design problems
  - Job sequencing and machine scheduling problems
  - Manpower planning problems
  - Production planning and distribution
  - Etc.

Combinatorial Optimization

- Combinatorial optimization problems are often **easy to state** but very **difficult to solve**.
- Many of the problems arising in applications are **NP-hard**, that is, it is strongly believed that they **cannot** be solved to optimality within **polynomially bounded computation time**.

Combinatorial Optimization

- Two classes of algorithms are available for the solution of combinatorial optimization problems:
  - Exact algorithms
  - Approximate algorithms

Combinatorial Optimization

- **Exact algorithms** are guaranteed to find the **optimal solution** and to prove its optimality for every finite size instance of a combinatorial optimization problem within an **instance-dependent run time**.
- In the case of NP-hard problems, in the worst case, **exponential time** to find the optimum.
- For most NP-hard problems the performance of exact algorithms **is not satisfactory**.

Combinatorial Optimization

- If optimal solutions cannot be efficiently obtained in practice, the only possibility is to trade optimality for efficiency.
- **Approximate algorithms**,
  - often also called **heuristic, hill climbing**, or **iterative improvement**
  - seek to obtain good, that is, near-optimal solutions at relatively low computational cost without being able to guarantee the optimality of solutions.

Hill Climbing

Hill Climbing

- General method to solve combinatorial optimization problems
- **Principle:**
  - Start with initial configuration
  - Repeatedly search neighborhood and select a neighbor as candidate
  - Evaluate some cost function (or fitness function) and accept candidate if "better"; if not, select another neighbor
  - Stop if quality is sufficiently high, if no improvement can be found or after some fixed time

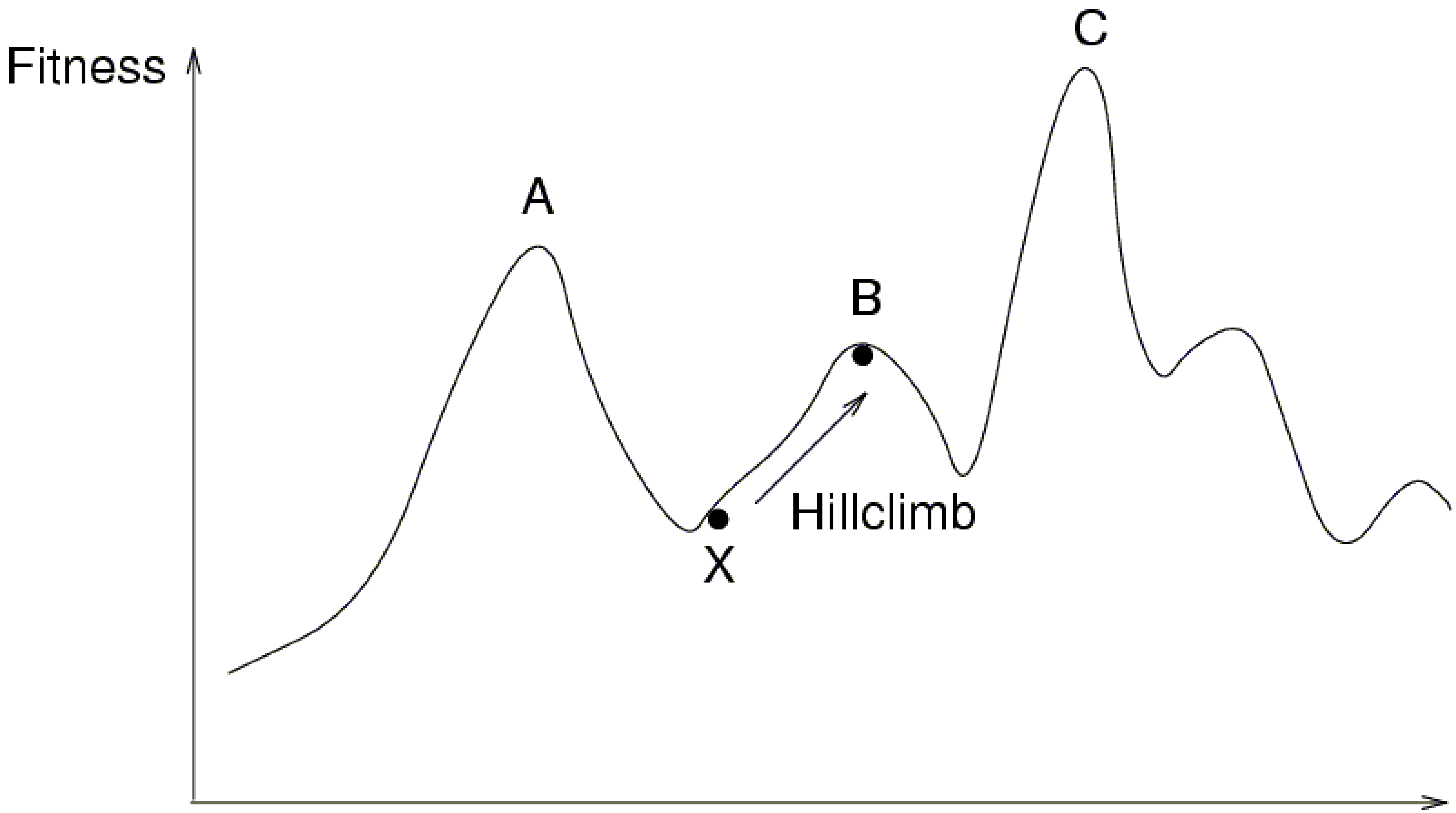
Hill Climbing

- Needed are:
  - A method to generate initial configuration
  - A transition or generation function to find a neighbor as next candidate
  - A cost function
  - An evaluation criterion
  - A stop criterion

Hill Climbing

- Hill Climbing:
  - Candidate is always and only accepted if cost is lower (or fitness is higher) than current configuration
  - Stop when no neighbor with lower cost (higher fitness) can be found
- Disadvantages:
  - Local optimum as best result
  - Local optimum depends on initial configuration
  - Generally no upper bound on iteration length

Hill climbing



How to cope with disadvantages

- Repeat algorithm many times with different initial configurations
- Use information gathered in previous runs
- Use a more complex Generation Function to jump out of local optimum
- Use a more complex Evaluation Criterion that accepts sometimes (randomly) also solutions away from the (local) optimum

Metaheuristics

Metaheuristics

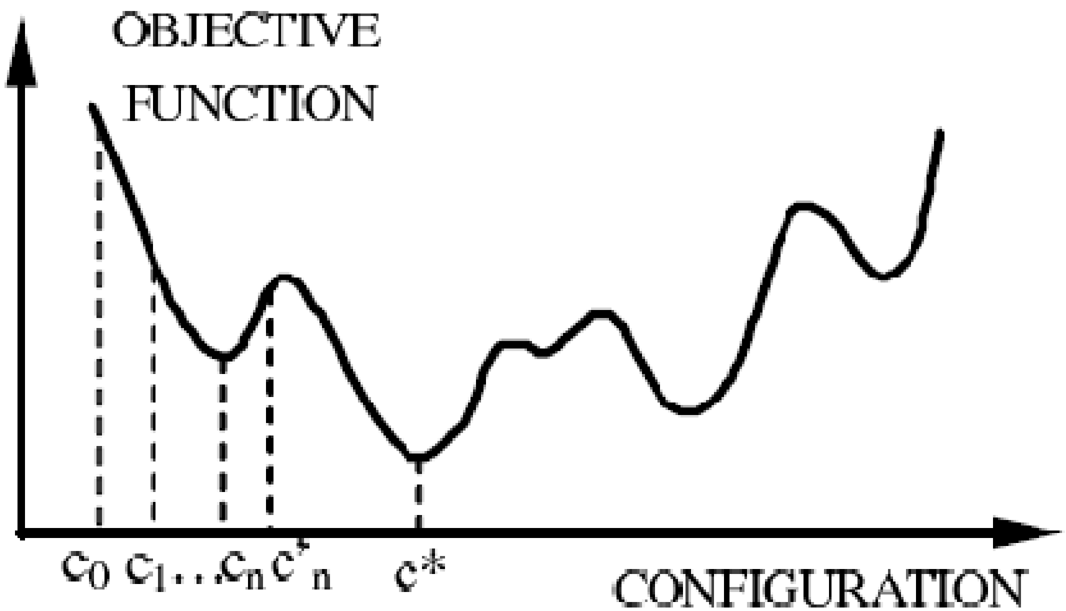
- A disadvantage of heuristic methods is that they:
  - either generate only a very limited number of different solutions, or
  - they stop at poor quality local optima, which is the case for iterative improvement methods.
- **Metaheuristics** have been proposed which try to bypass these problems.
- Metaheuristics apply to solve the problems known as of **difficult optimization**
- Available from the 1980s

Metaheuristics

- **Definition**:
  - A **metaheuristic** is a set of algorithmic concepts that can be used to define heuristic methods applicable to **a wide set of** different problems.
  - A **metaheuristic** can be seen as a **general-purpose heuristic method** toward promising regions of the search space containing high-quality solutions.
  - A metaheuristic is a general algorithmic framework which can be applied to different optimization problems with relatively **few modifications** to make them adapted to a specific problem.

Capability of Metaheuristics

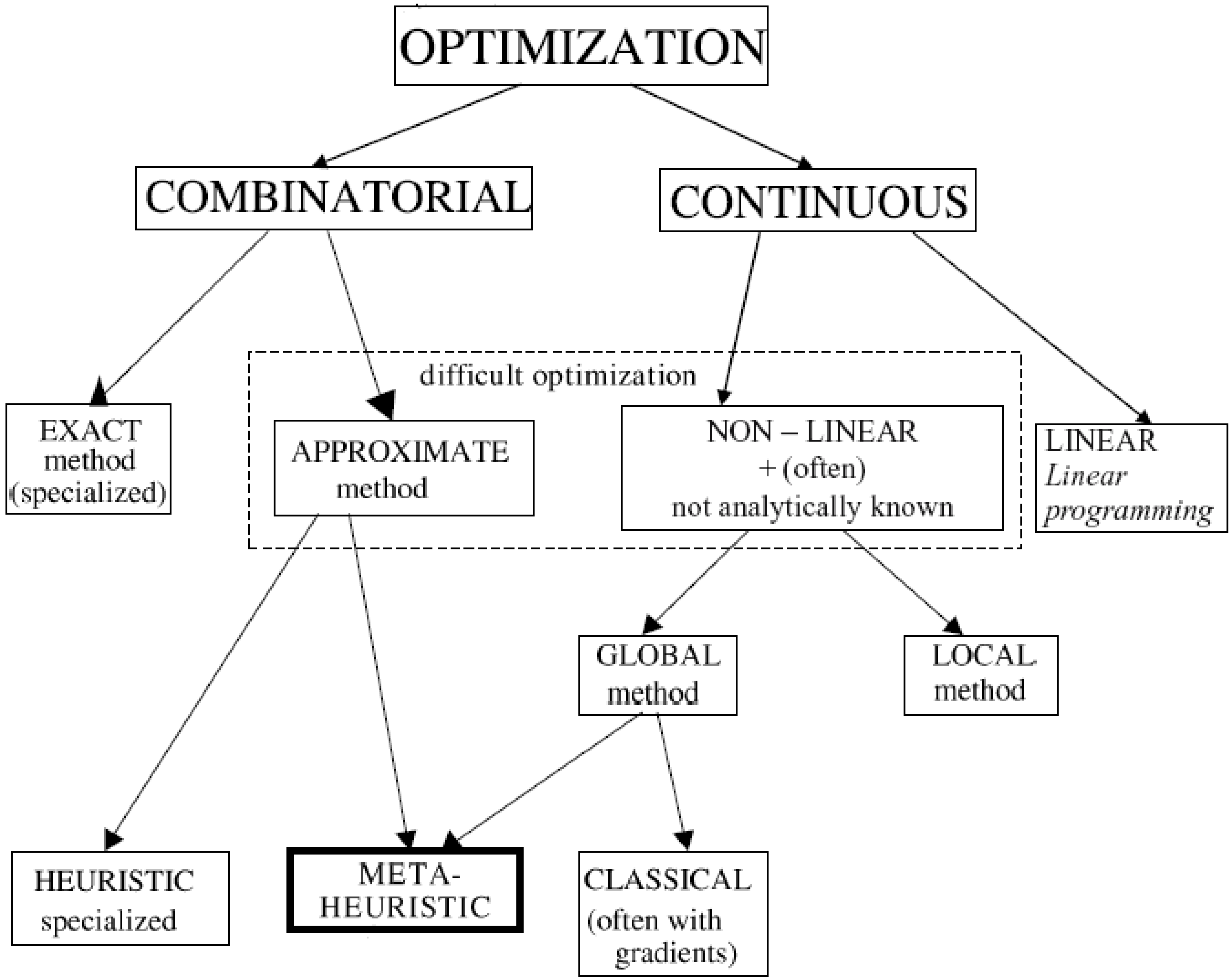
- **Metaheuristics have capability to be extracted from a local minimum**



Metaheuristics

- The metaheuristics are from now on regularly employed in all the sectors of engineering,
- Examples of metaheuristics algorithms:
  - The evolutionary algorithms
  - The tabu search method
  - The ant colony optimization
  - The simulated annealing method
  - Etc.

Metaheuristics



References

References

- J. Dreo A. Petrowski, P. Siarry E. Taillard, **Metaheuristics for Hard Optimization**, Springer-Verlag, 2006.
- R.J. Moraga, G.W. DePuy, G.E. Whitehouse, **Metaheuristics: A Solution Methodology for Optimization Problems**, Handbook of Industrial and Systems Engineering, A.B. Badiru (Ed.), 2006.

The End