Artificial Intelligence Methods (G52AIM)

Dr Rong Qu
rxq@cs.nott.ac.uk

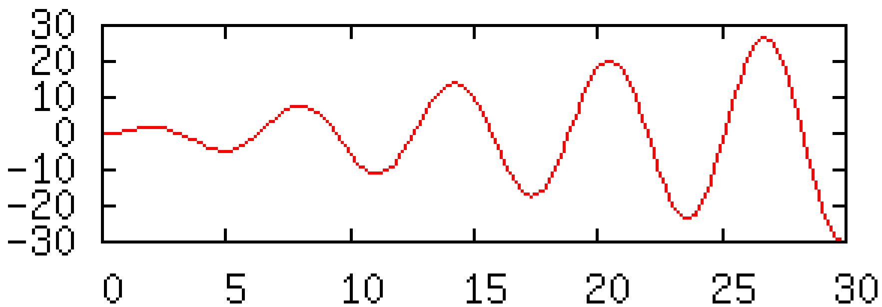
Local Search

Optimisation Problems: Definition

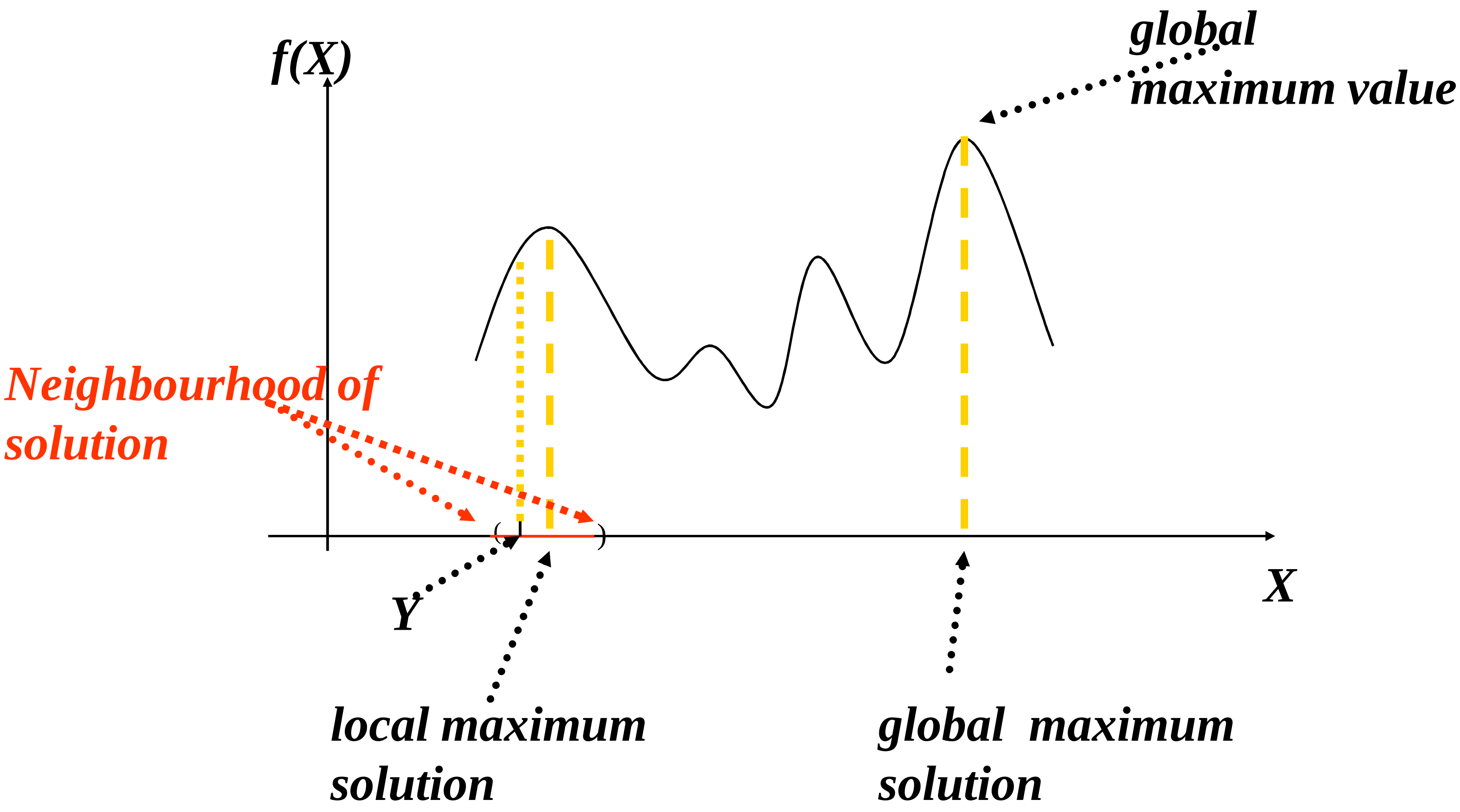
- Find values of a given set of decision variables: $X=(x_1, x_2, \ldots, x_n)$ which maximises (or minimises) the value of an objective function: $x_0 = f(x_1, x_2, \ldots, x_n)$, subject to a set of constraints

- Any vector X, which satisfies the constraints is called a **feasible solution** and among them, the one which maximise (or minimise) the objective function is called the **optimal solution**

Maximise: $x \sin(x)$
subject to $0 \leq x \leq 30$

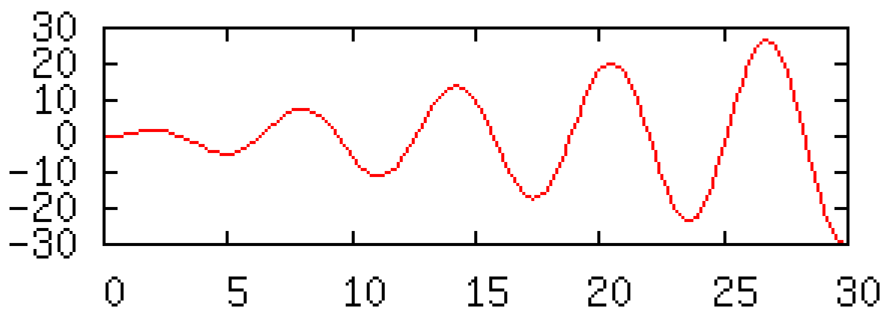


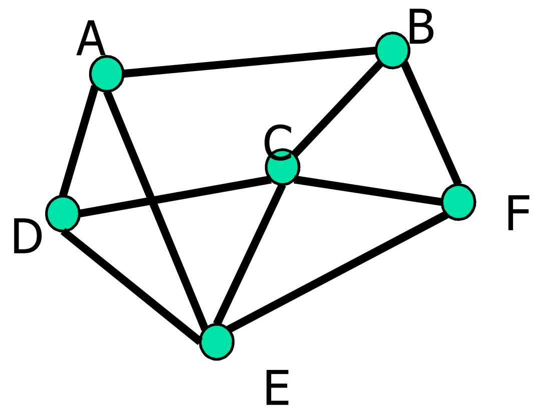
Optimisation Problems: terminology



Optimisation Problems: Difficulties

- For most of real world problems
  - An exact model cannot be built easily;
  - Number of feasible solutions grow exponentially with growth in the size of the problem.



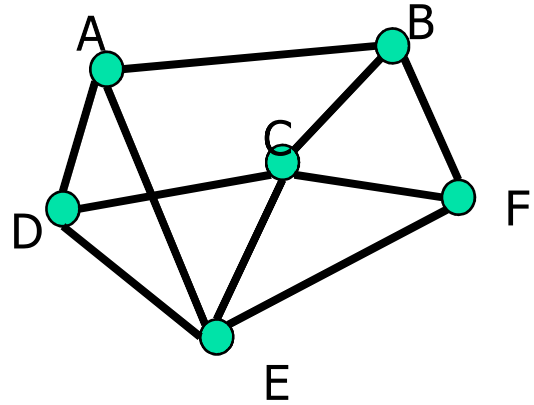


Methods of optimisation

- Mathematical optimisation
  - Based on Mathematical techniques to solve the optimisation problem exactly or approximately with guarantee for quality of the solution;
  - Examples:
    - Simplex method: by far the worlds most widely used optimization algorithm!
    - Lagrange multipliers, branch and bound, cutting planes, interior point methods, etc;

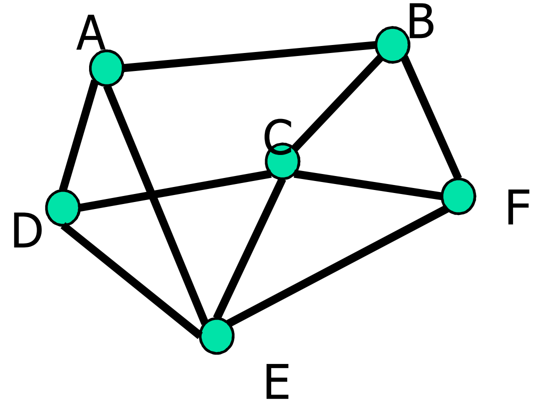
Methods of optimisation

- Mathematical optimisation
  - +: Guarantee of optimality
  - - : Unable to solve larger instances of difficult problems due to large amount of computational time and memory needed;



Methods of optimisation

- Constructive Heuristics
  - Using simple minded greedy functions to evaluate different options (choices) to build a reasonable solution iteratively (one element at a time);
  - Examples: Dijkastra method, Big M, Two phase method, Density constructive methods for clustering problems, etc;

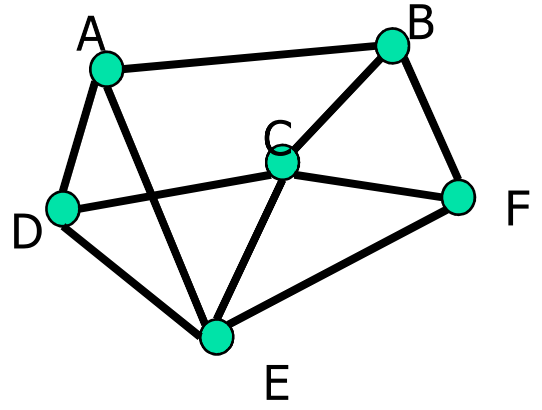


Methods of optimisation

- Constructive Heuristics
  - +: Ease of implementation;
  - -: Poor quality of solution;
  - -: Problem specific.

Methods of optimisation (cont.)

- Local Search algorithms
  - A neighbourhood search or so called local search method starts from some **initial (complete) solution** and moves to a better neighbouring solution until it arrives at a **local optimum**, one that does not have a better neighbour.
  - Examples: *k-opt* for TSP, etc;



Methods of optimisation (cont.)

- Local Search algorithms
  - +: Ease of implementation;
  - +: Guarantee of local optimality usually in small computational time;
  - +: No need for exact model of the problem;
  - -: Poor quality of solution due to getting stuck in poor local optima;

Methods in G52AIM

- Meta-heuristics
  - These algorithms guide an underlying heuristic / local search to escape from being trapped in a local optima and to explore better areas of the solution space;

Methods in G52AIM

- Meta-heuristics
  - Local search based approaches
    - Hill climbing
      - “Run uphill/downhill and hope you find the top/bottom”
    - Simulated annealing
      - “Shake it up a lot and then slowly let it settle”
    - Tabu search
      - “Don’t look under the same lamp-post twice”
    - GRASP, VNS, etc

Methods in G52AIM

- Meta-heuristics
  - Population based approach
    - Genetic algorithms
      - "survival of the fittest"
    - Ant algorithms
      - "wander around a lot and leave a trail"
    - Genetic programming
      - Learn to program
    - Evolutionary algorithms, etc

Methods in G52AIM

- Meta-heuristics
  - +: Able to cope with inaccuracies of data and model, large sizes of the problem and real-time problem solving;
  - +: Including mechanisms to escape from local optima of their embedded local search algorithms,
  - +: Ease of implementation;
  - +: No need for exact model of the problem;
  - -: Usually no guarantee of optimality.

Problems in G52AIM

- Problem domain: optimiation
  - Quality of solutions: given by evaluation function (objective function, fitness, etc)
  - Aim: minimize or maximize this objective

Local Search Methods

- Initially focus on algorithms that might be classed as "iterative improvement"
- Take a candidate complete 'solution' and then try to fix or repair it
- Simplest version of this is "local search"

Local Search Methods

- A **neighbourhood function** is usually defined by using the concept of a move, which changes one or more attributes of a given solution to generate another solution.
- Definition
  - A solution x is called a local optimum with respect to the neighbourhood function N, if $f(x) < f(y)$ for every y in $N(x)$.

Local Search Methods

- Why local search?
  - Exponential growth of the solution space for most of the practical problems;
  - Ambiguity of the model of the problem for being solved with exact algorithms;
  - Ease of use of problem specific knowledge in design of algorithm than in design of classical optimisation methods for a specific problem.

Local Search Methods

- Elements
  - Representation of the solution;
  - Evaluation function;
  - Neighbourhood function
    - To define solutions which can be considered close to a given solution.
    - For example: For optimisation of real-valued functions in elementary calculus, for a current solution *x0*, neighbourhood is defined as an interval *(x0–r, x0+r)*.

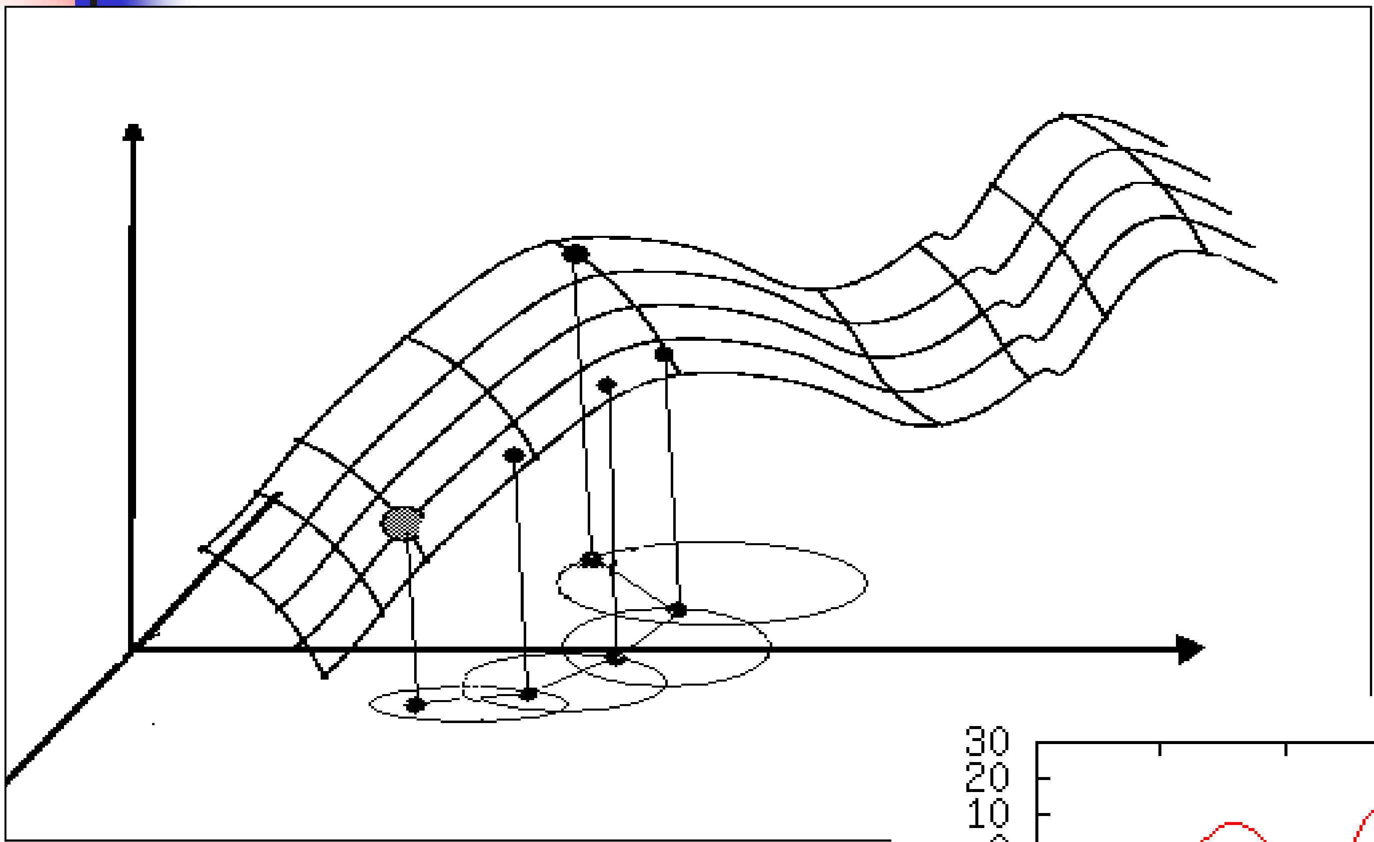
Local Search Methods

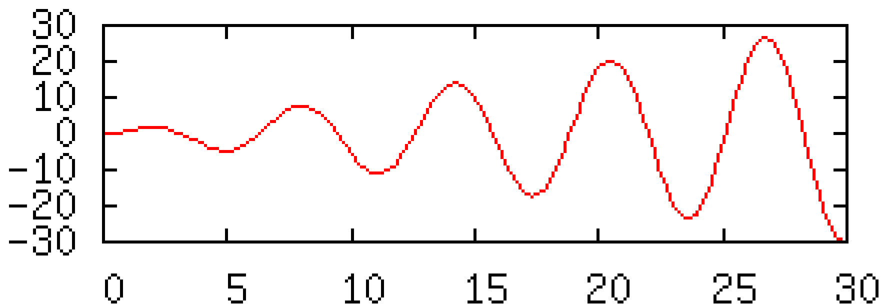
- Elements
  - Neighbourhood search strategy
    - Random and systematic search;
  - Acceptance criterion
    - first improvement; best improvement; best of non-improving solutions; random criteria

Hill Climbing - Algorithm

1. Pick a random point in the search space
2. Consider all the neighbours of the current state
3. Choose the neighbour with the best quality and move to that state
4. Repeat 2 thru 4 until all the neighbouring states are of lower quality
5. Return the current state as the solution state

Hill Climbing - Examples





Hill Climbing – Exercise

- Problem
  - TSP: A travelling salesman is visiting *n* cities
  - Constraint: He can visit each city only once, and back to the starting city
  - Objective: Find the shortest tour
- Task
  - Apply the hill climbing method to solve TSP

Hill Climbing – Exercise

- How do we define the problem as a search problem?
  - What is a "point" in the **search space**?
  - How to represent a solution?
  - What is the search space?

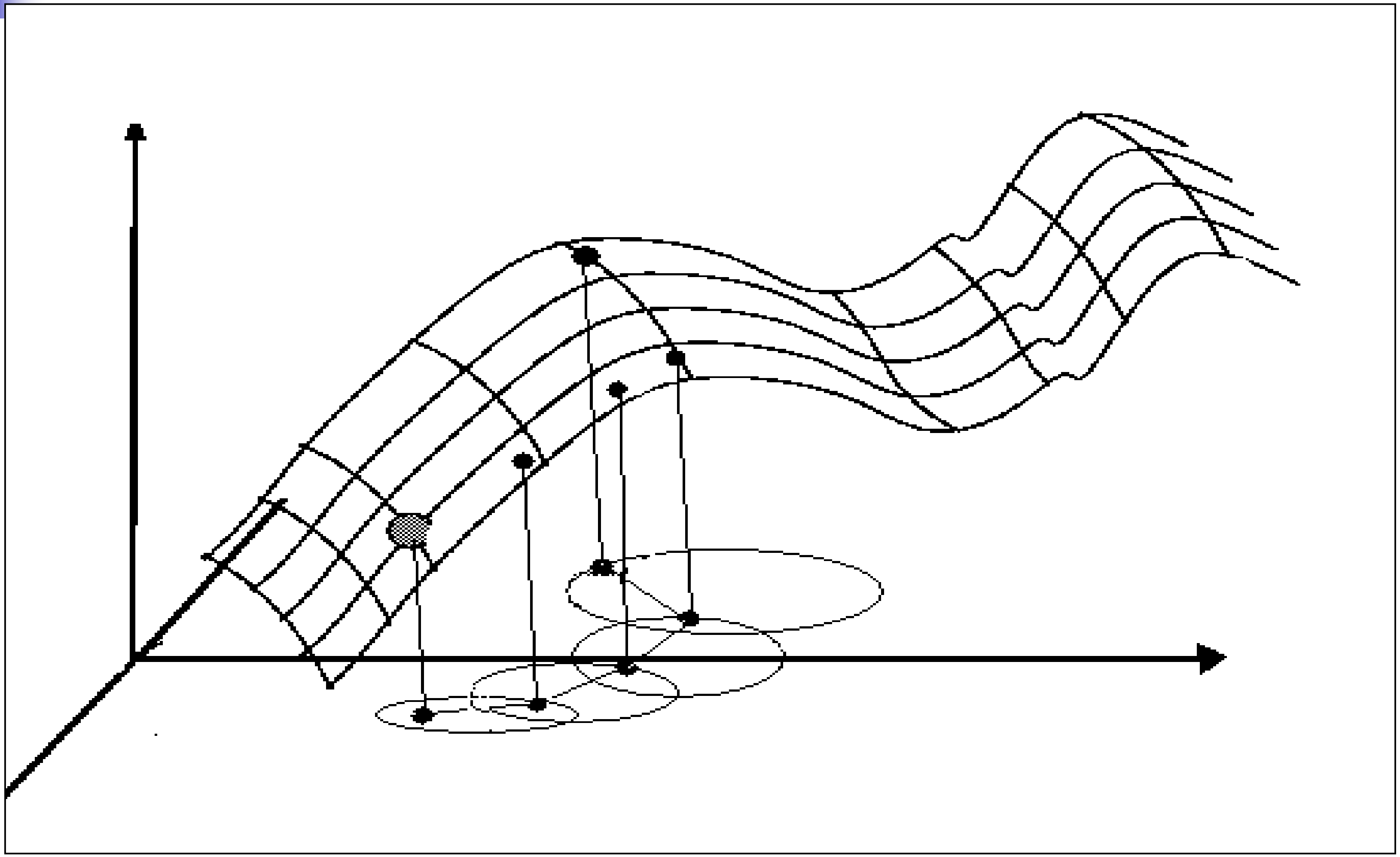
Hill Climbing – Exercise

- How do we design the HC for TSP?
  - What is your evaluation function?
  - What is your neighbour?
  - What is your search strategy?
  - What is your acceptance criteria?

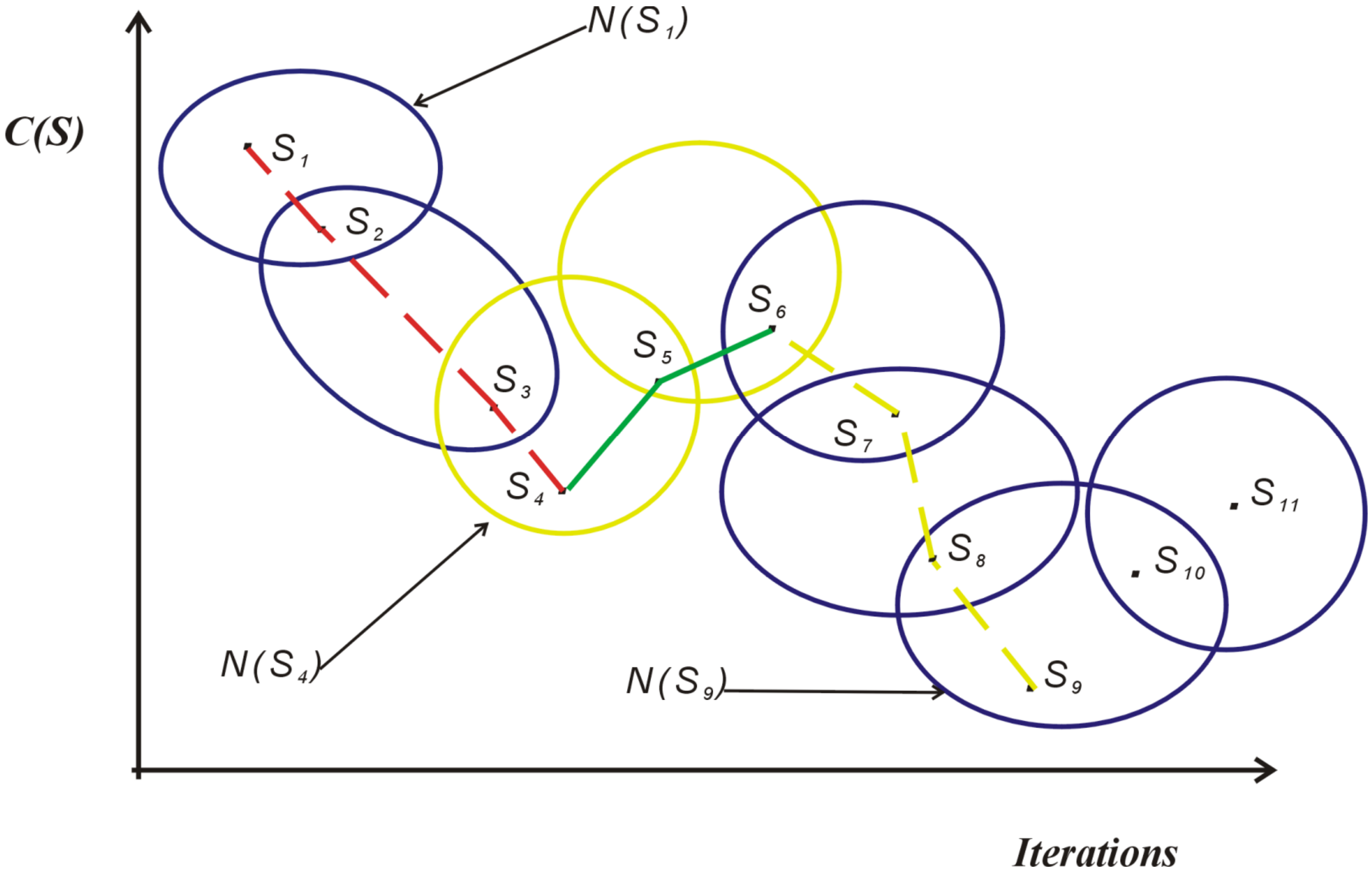
Hill Climbing – Exercise

- What's the problem in HC?
  - First improvement
    - Within the neighbourhood, if a point x' is an improvement then immediately terminate and jump straight to it
  - Best Improvement
    - Search the entire neighbourhood, and find the x' that gives the largest improvement
    - What to do in tie-breaks?

Hill Climbing – local optima



How can bad local optima be avoided?



Summary

- Optimisation problems
  - Definition
  - Methods
- Local search algorithms
  - Elements
  - Hill climbing

Learning objectives

- Terminology
  - Local vs. global optima
  - Neighborhood
  - Feasibility
- Local search algorithms
  - Concepts
  - Elements
  - Basic hill climbing