Cooperative and Adaptive Algorithms

Instructor: Alaa Khamis, Ph.D, SMIEEE

Senior Research Scientist vestec

TAs: Sepehr Eghbali (ECE)

Keyvan Golestan Irani (ECE

Course Presentation – Tuesday May 6, 2014

Outline

- Course Description
- Course Objectives
- Course Topics
- Course Outline
- Teaching Methodology
- Course Policy
- Resources

Outline

- **Course Description**
- Course Objectives
- Course Topics
- Course Outline
- Teaching Methodology
- Course Policy
- Resources

Course Description

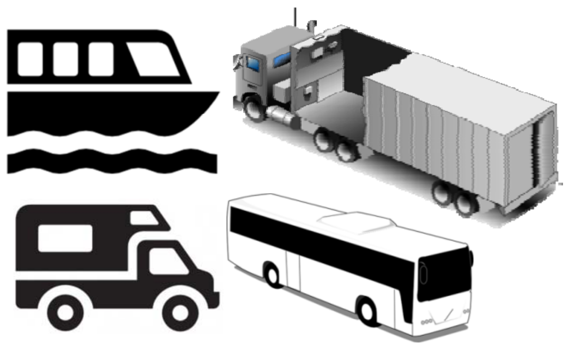
The course addresses the **ill-structured problems** and need for computational intelligence methods.

The course introduces the concepts of **heuristics** and their use in conjunction with search methods, solving problems using heuristics and metaheuristics.

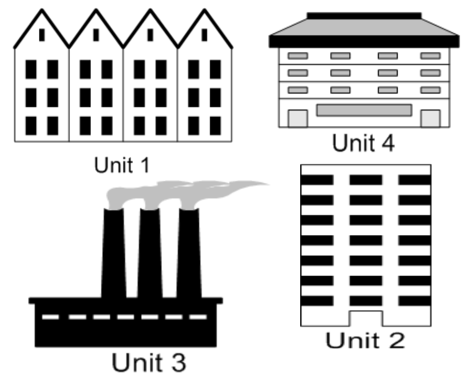
The course also introduces the concepts of **cooperation and adaptation** and how they are influencing new methods for solving **complex problems**.

Course Description

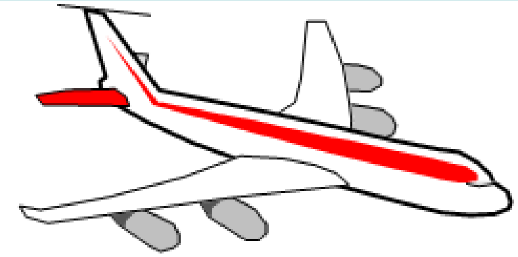
- Complex problems are everywhere



Vehicle routing

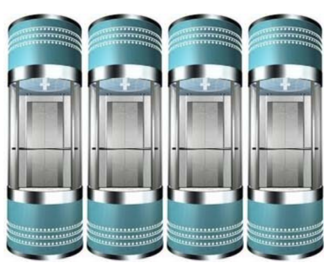


Plant layout

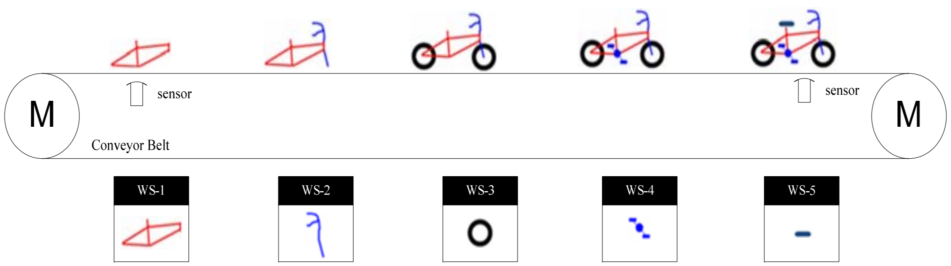


Airline scheduling

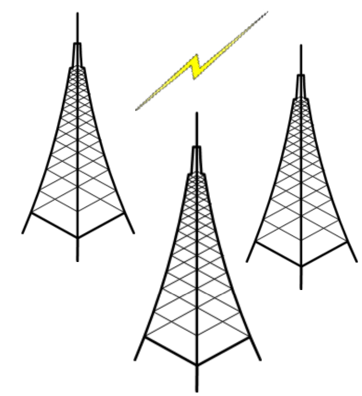
Railway scheduling



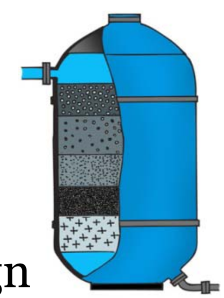
Elevator dispatching



Assembly line balancing

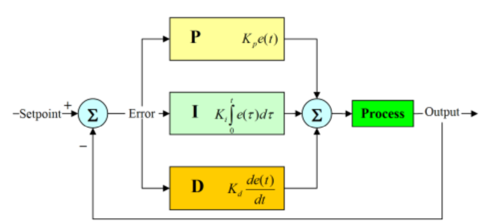


Cell assignment



Filter design

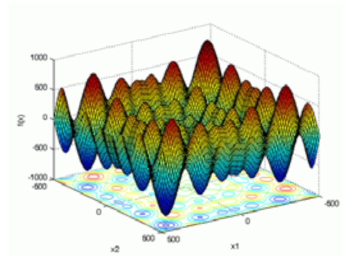
Path Planning



PID Auto-Tuning

Course Description

- Some Sources of Complexity
  - Multimodality



  - Lack of exact mathematical model

$$\min \quad \sum_{i,j} c_{ij} x_{ij}$$

$$\sum_i x_{ij} = 1 \quad \forall i, \qquad \sum_j x_{ij} = 1 \quad \forall j,$$

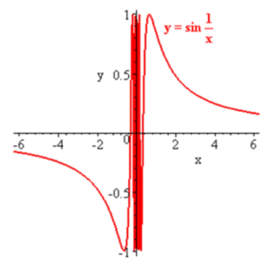
$$0 \le x_{ij} \le 1 \quad x_{ij} \text{ integer},$$

$$u_1 = 1,$$

$$2 \le u_i \le n \quad \forall i \ne 1,$$

$$u_i - u_j + 1 \le (n-1)(1 - x_{ij}) \quad \forall i \ne 1, \forall j \ne 1.$$

  - Non-Differentiability



  - Combinatorial nature

  - Distributed nature

Course Description

Engineering Problems

well-structured Problems

- Can be stated in terms of numerical variables.
- Goals can be specified in terms of a well defined objective function.
- There exists an algorithmic solution.
- Correct answers required

ill-Structured Problems[1]

- Those which do not meet one or more of the left.
- Division between well and ill-structured is less precise.
- Some problem even they may be well structured in principle, they may be ill-structured in practice [2].

Course Description

- **Solution strategies for ill-structured problems**
  - ◇ Retrieving a solution or an answer.
  - ◇ Starting from a guess and then improve on it .
  - ◇ Searching among alternatives.
  - ◇ Searching forward from the problem to the answer.
  - ◇ Searching backward from a goal to the situations of the problem.

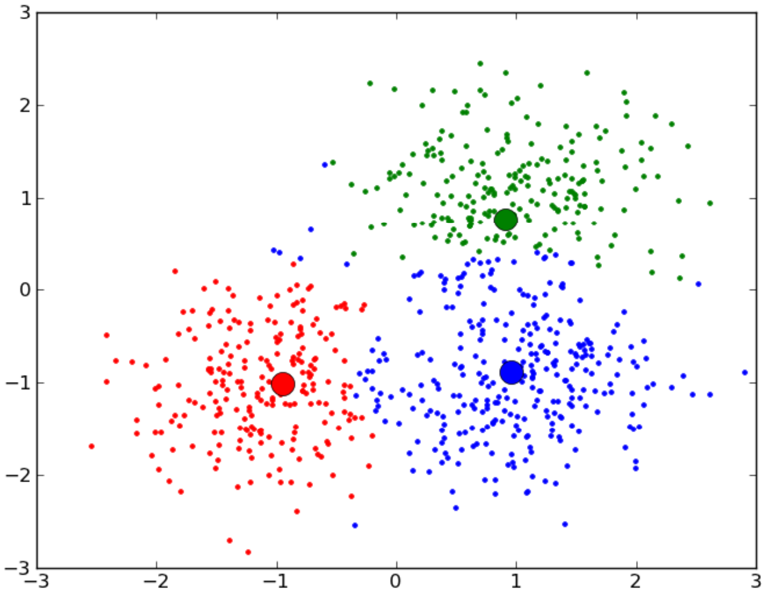
[3]

Course Description

- **Example of well-structured problem but practically ill-structured**

The Clustering Problem

◇ Given ***n*** objects, group them in ***c*** groups (clusters) in such a way that all objects in a single group have a “natural" relation to one another, and objects not in the same group are somehow different.

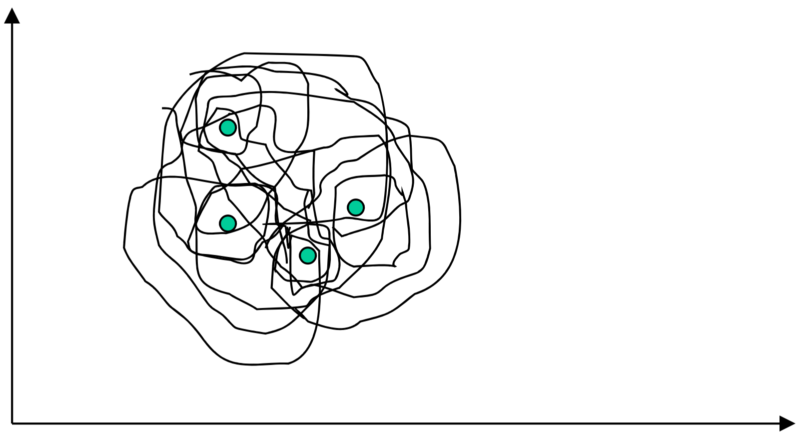


[3]

Course Description

- **Example of well-structured problem but practically ill-structured**

The Clustering Problem



[3]

Course Description

- **Example of well-structured problem but practically ill-structured**

The Clustering Problem: Exponential Growth

$$N(n,c)=\left(\frac{1}{c!}\right)\sum_{m=1}^{c}(-1)^{c-m}\binom{c}{m}m^{n}$$

[See partition theory]

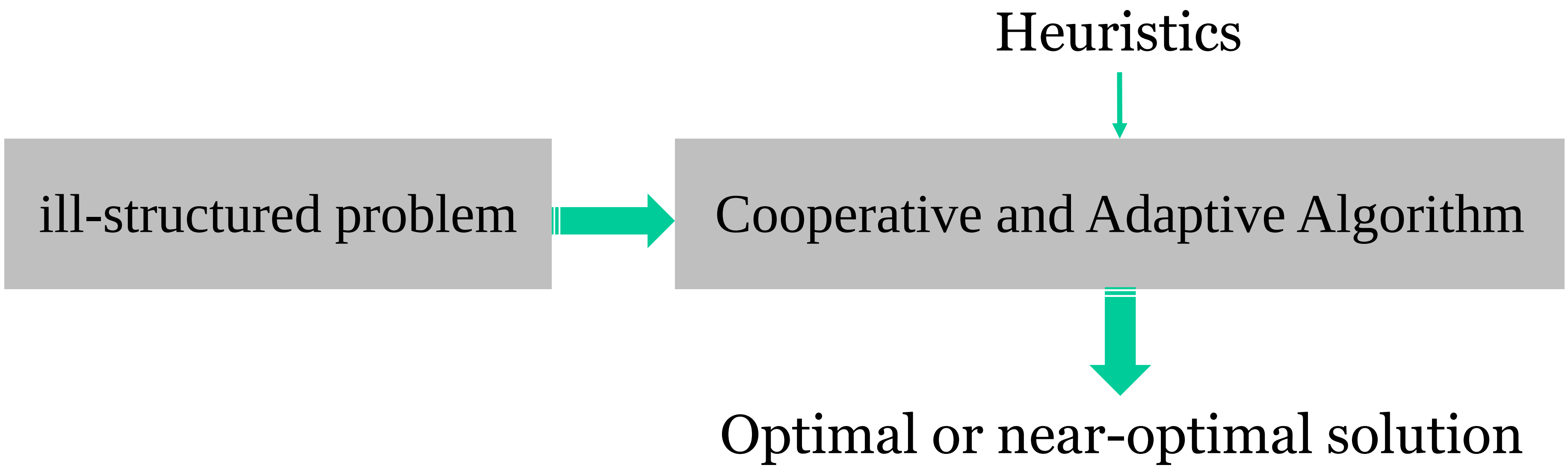
◇ Example: If $\boldsymbol{n=50}$ and $\mathbf{c=4}$, the number is $\mathbf{5.3x10^{28}}$

◇ If $\boldsymbol{n}$ is increase to $\mathbf{100}$, the number becomes $\mathbf{6.7x10^{58}}$.

◇ So enumerating all possible partitions for large problems is practically not feasible

[3]

Course Description



Course Description

- **Adaptive Algorithm**
  - ◇ To adjust to new or different situations
  - ◇ Ability to improve behaviour
  - ◇ Evolution
  - ◇ Learning from instructor, example or by discovery
  - ◇ Generalization

[3]

Course Description

- **Adaptive Algorithm: Types of adaptation**

  ◇ **Deterministic:** when the control parameter is changed according to some deterministic update rule without taking into account any information from the search algorithm,

  ◇ **Adaptive:** when the update rule takes information from the search algorithm and changes the control parameter accordingly,

  ◇ **Self-adaptive:** when the update rule itself is being adapted.

[5]

Course Description

- **Cooperative Algorithm**
  - ◊ A group to solve a joint problem or perform a common task(s) based on sharing the responsibility for reaching the goal.
  - ◊ Direct cooperation
  - ◊ Indirect/Stigmergic cooperation
  - ◊ Independent actions

[3]

Course Description

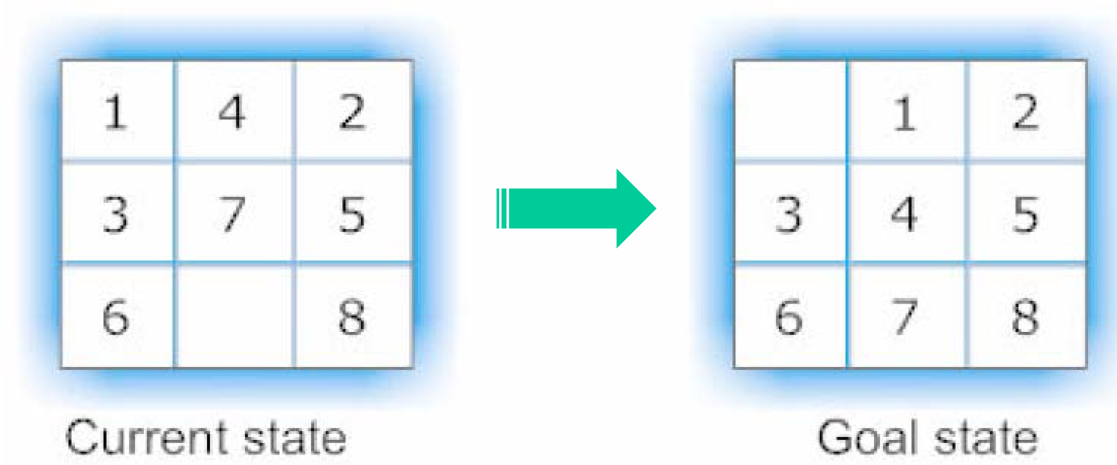
- **Heuristics**

Heuristics is a solution strategy or rules by **trial-and-error** to produce acceptable (**optimal or near-optimal**) solutions to a complex problem in a reasonably practical time.

Examples: FIFO, LIFO, earliest due date first, largest processing time first, shortest distance first, etc.

Course Description

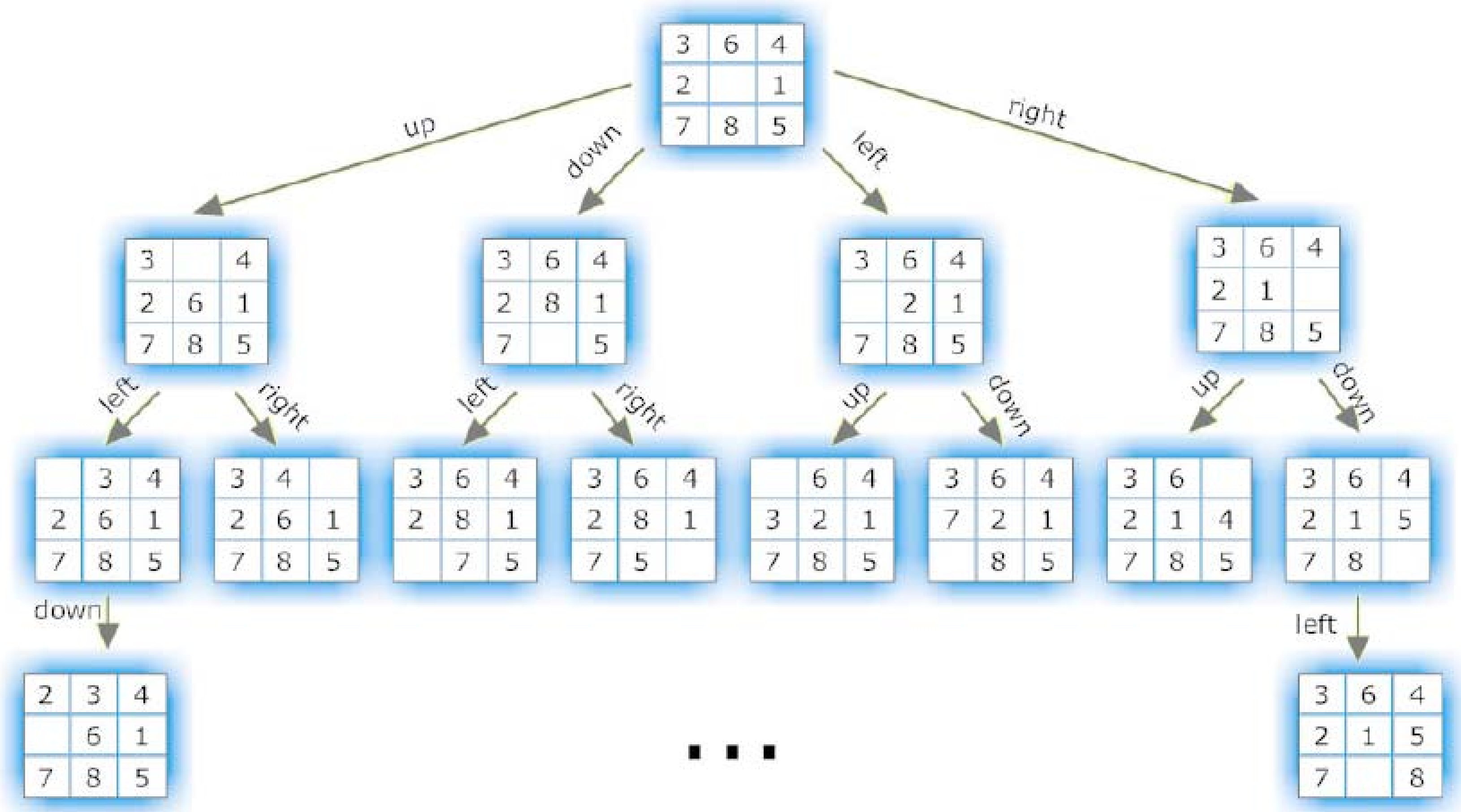
- **Heuristics:** 8-Puzzle/sliding-block puzzle/tile-puzzle Problem



[4]

Course Description

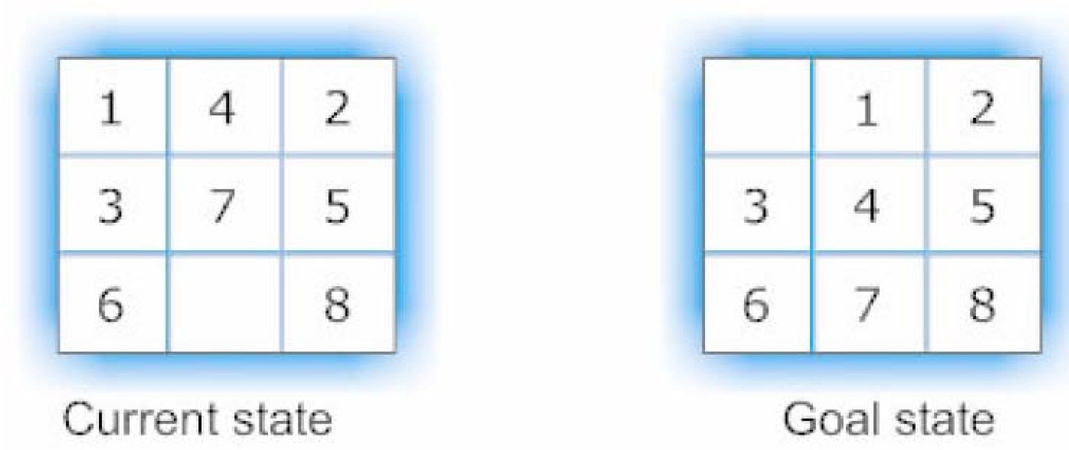
- **Heuristics:** 8-Puzzle/sliding-block puzzle/tile-puzzle Problem



The 8-puzzle can be solved in 31 moves or fewer. [4]

Course Description

- **Heuristics:** 8-Puzzle/sliding-block puzzle/tile-puzzle Problem



◇ We can define the following heuristics :

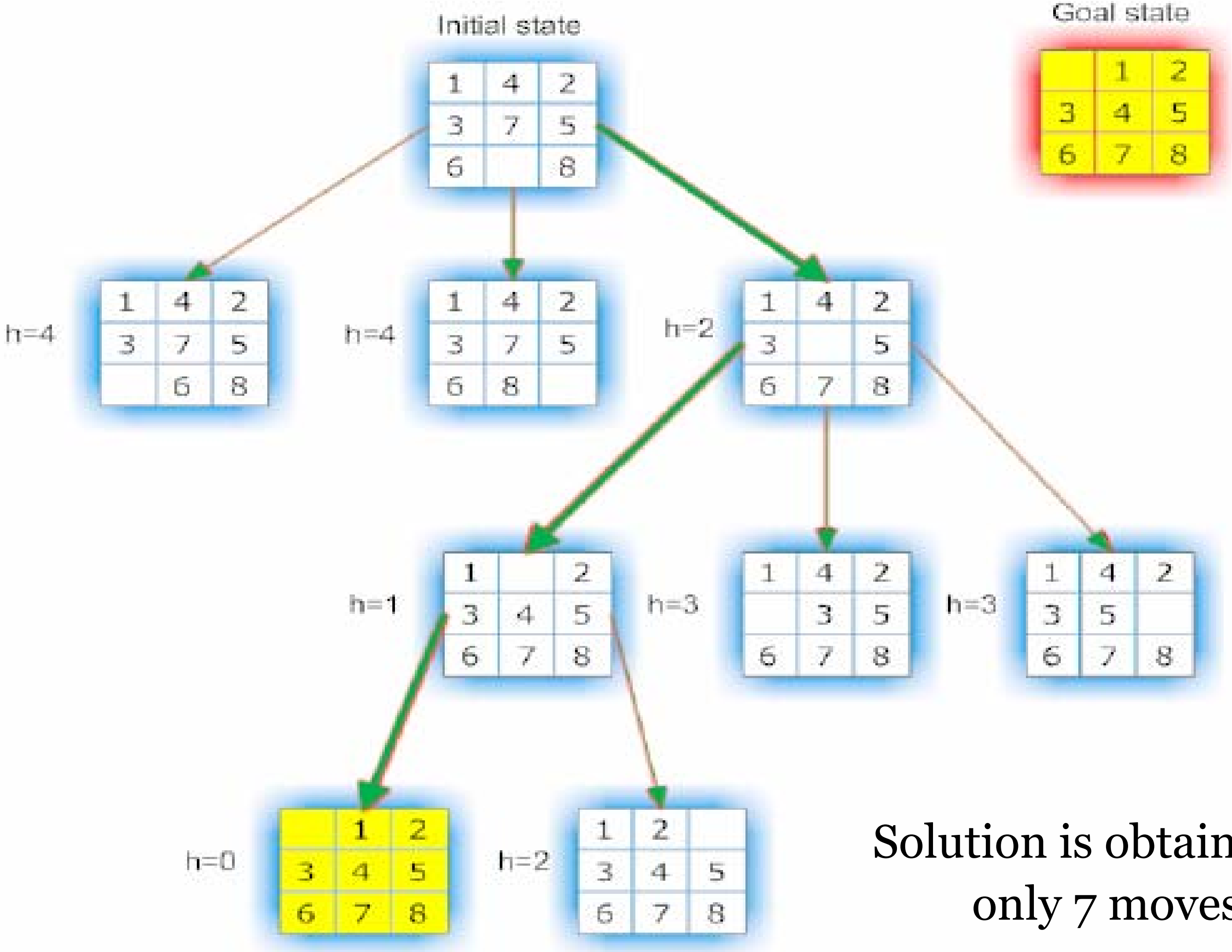
◇ $h_1(n)$ = the number of **misplaced tiles** in the current state relative to the goal state.

◇ In this example, $h_1(n)$= 4 because tiles 1, 4, 7 and blank are out of place. Obviously, lower values for h_1 are preferred.

[4]

Meta-Heuristic Optimization

- **Meta-Heuristics: 8-Puzzle Problem**



Solution is obtained in only 7 moves

[4]

Outline

- Course Description
- **Course Objectives**
- Course Topics
- Course Outline
- Teaching Methodology
- Course Policy
- Resources

Course Objectives

- Introduce the concepts of **cooperation** and **adaptations** and how they are influencing new methods for **solving complex problems**.
- Study **meta-heuristics**, **evolutionary computing methods**, **swarm intelligence**, **ant-colony algorithms**, **particle swarm methods**.
- Illustrate the use of these algorithms in solving **continuous and discrete problems** that arise in engineering applications.

Outline

- Course Description
- Course Objectives
- **<u>Course Topics</u>**
- Course Outline
- Teaching Methodology
- Course Policy
- Resources

Course Topics

Optimization techniques are **search methods**, where the goal is to find a solution to an optimization problem, such that a given quantity is optimized, possibly subject to a set of constraints.

Optimization Algorithms

Exact Algorithms

- Find the optimal solution
- High computational costs.

Approximate Algorithms

- Find near-optimal solutions
- Low computational costs.

In most of the real applications, **quickly finding near-optimal** solution is better than **spending large amount of time** in search for optimal solution.

Course Topics

Search Methods

- Deterministic
  - Numerical Methods
  - **Graph Search Methods**
    - **Blind Search**
      - ◇ Depth-first
      - ◇ Breadth-first
      - ◇ Dijkstra
    - **Informed Search**
      - ◇ Hill climbing
      - ◇ Beam search
      - ◇ Best-first, A*
    - **Game Playing as search**
- Stochastic
  - **Trajectory Methods**
    - ◇ **TS**
    - ◇ **SA**
    - ◇ GRASP
    - ◇ ILS
    - ◇ GLS
    - ◇ VNS
  - **Population-based Methods**
    - Evolutionary Computing
      - ◇ **GA**
      - ◇ ES
      - ◇ EP
      - ◇ GP
    - Swarm Intelligence
      - ◇ **PSO**
      - ◇ **ACO**
      - ◇ ABC, FA, HS, SDS

Note: Topics highlighted in **bold** will be covered in this course.

Course Topics

Introductory Concepts
• Optimization Theory and Combinatorics • Graph-search Algorithms • Game Playing as Search • Metaheuristics
Trajectory-based Metaheuristic Algorithms
• Tabu Search • Simulated Annealing
Population-based Metaheuristic Algorithms
• P-Metaheuristics • Evolutionary Computation • Genetic Algorithms • Swarm Intelligence • Particle Swarm Optimization • Ant Colony Optimization

Outline

- Course Description
- Course Objectives
- Course Topics
- **Course Outline**
- Teaching Methodology
- Course Policy
- Resources

Course Outline

1. Optimization Theory and Combinatorics
2. Graph Search Algorithms
3. Game Playing as Search
4. Trajectory-based Metaheuristic Optimization
5. Tabu Search
6. Simulated Annealing
7. Population-based Metaheuristic Optimization
8. Evolutionary Computation
9. Genetic Algorithms
10. Swarm Intelligence
11. Particle Swarm Optimization (PSO)
12. Ant Colony Optimization (ACO)

Outline

- Course Description
- Course Objectives
- Course Topics
- Course Outline
- **Teaching Methodology**
- Course Policy
- Resources

Teaching Methodology

Algorithm biological/non-biological source of inspiration

Algorithm psuedo-code

Algorithm heuristics, parameters, cooperation and adaptation

Algorithm pros and cons

Algorithm applications

Discrete problems

Continuous problems

Hand-iterations for small sub-set of the problem

Programming for real problem

Algorithm refinement and tuning

Outline

- Course Description
- Course Objectives
- Course Topics
- Course Outline
- Teaching Methodology
- **Course Policy**
- Resources

Course Policy

Evaluation Method	Weight
Assignments	20%
Projects	30%
Final Exam	50%

Course Resources

- **Course Website**

  Lecture slides, tutorials, assignments, code will be posted on the course webpage:

  http://www.alaakhamis.org/teaching/ECE457A/index.html

Course Resources

- **Textbook (Optional)**
  - ◇ A. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. Wiley, 2006.
  - ◇ Xin-SheYang. *Engineering Optimization: An Introduction with Metaheuristic Applications*. A JOHN WILEY & SONS, INC., 2010.
  - ◇ Singiresu S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, INC., 2009.
  - ◇ C. Revees. *Modern heuristic techniques for combinatorial problems*. Halsted Press, New York, 1993.

Course Resources

- **Books put on Reserve in the Library**
  - ◇ Andries P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley and Sons, 2006.
  - ◇ James F. Kennedy, Russell C. Eberhart. *Swarm Intelligence*. Yuhui. Shi Published 2001 by Morgan Kaufmann
  - ◇ Eiben and A, Smith. *Introduction to Evolutionary Computing*. J.E, Springer, Berlin, 2003.
  - ◇ Revees, C. *Modern heuristic techniques for combinatorial problems*. Halsted Press, New York, 1993.

References

1. Newell. Heuristic Programming: Ill-Structured Problems. Book Chapter in Progress in Operation Research, Vol. III, Aronofsky, ed. – 1969.
2. Simon, H.A. (1973), “Structure of ill structured problems”, Artificial Intelligence, Vol. 4 Nos 3-4, pp. 181-201.
3. M. Kamel, ECE457A: Cooperative and Adaptive Algorithms. Spring 2011, University of Waterloo.
4. Crina Grosan and Ajith Abraham. *Intelligent Systems: A Modern Approach*. Springer, 2011.
5. A. E. Eiben, R. Hinterding and Z. Michaleweiz.”Parameter Control in Evolutionary Algorithms”. IEEE Transactions on Evolutionary Computing, vol. 3, issue 2, pp. 124-141, 1999.

Questions?