SPECIAL EDITION FOR CSEDU STUDENTS

TOUCH-N-PASS EXAM CRAM GUIDE SERIES

ARTIFICIAL INTELLIGENCE

Prepared By

Sharafat Ibn Mollah Mosharraf

CSE, DU
12th Batch (2005-2006)

Table of Contents

CHAPTER 2
INTELLIGENT AGENTS

2.1	Suppose you have developed a program to solve geometric analysis problems that appear in IQ tests. In several occasions it is found that your program responds more efficiently and intelligently with respect to human response. Would we then conclude that a program is more intelligent than human? Explain. [*2003. Marks: 4*]
2.2	What do you understand by intelligent machine and intelligent agent? Is computer an intelligent machine? Justify your answer. [2008, *2007. Marks: 3*]
2.3	What is an agent? Describe the structure of intelligent agent. [*2003. Marks: 2 + 4*]
2.4	Define with example the following terms: [*2007, 2004. Marks: 0.5 to 1 each*] 1. Agent function 2. Agent program 3. Rationality 4. Learning agent 5. State 6. State space 7. Search tree 8. Search node 9. Goal 10. Action 11. Successor function 12. Branching factor.
2.5	Distinguish between agent functions and agent programs. [*In-course. Marks: 3*]
2.6	Both the performance measure and the utility function measure how well an agent is doing. Explain the difference between the two. [*In-course. Marks: 4*]
2.7	Explain the function of learning agent. [*2005. Marks: 4*]
2.8	What do you mean by PEAS? [*2005. Marks: 3*]
2.9	Give the structure of a simple reflex agent and the agent program for that. [*2005, 2004. Marks: 5*]
2.10	For each of the following agents, develop a PEAS description of the task environment: i. Robot Soccer player ii. Automated taxi driving agent iii. Internet book-shopping agent [*2007. Marks: 6*]

CHAPTER 3
SOLVING PROBLEMS BY SEARCHING

3.1	**What is agent? Characterize a general problem solving agent. [*2007. Marks: 3*]**
3.2	**How to define a problem formally? Explain why problem formulation must follow goal formulation. [*2007, 2005, 2004, 2003. Marks: 4*]**
3.3	**Prove that breadth-first search is a special case of uniform-cost search. [*2008, 2006. Marks: 3*]**
3.4	**Prove that uniform cost-search and breadth-first search with constant step costs are optimal when used with the graph-search algorithm. [*2007. Marks: 4*]**
3.5	**Describe a state space in which iterative deepening search performs much worse than depth-first search (for example, $O(n^2)$ vs. $O(n)$). [*2007. Marks: 4*]**
3.6	**Consider a state space where the start state is number 1 and the successor function for state n returns two states, number $2n$ and $2n + 1$.** **i. Draw the portion of the state space for states 1 to 15.** **ii. Suppose the goal state is 11. List the order in which nodes will be visited for breadth-first search, depth-limited search with limit 3, and iterative deepening search. [*2007. Marks: 4*]**
3.7	**Consider a state space where the start state is number 1 and the successor function for state n returns two states, number $2n$ and $2n + 1$.** **i. Draw the portion of the state space for states 1 to 35.** **ii. Suppose the goal state is 21. List the order in which nodes will be visited for breadth-first search, depth-first search, depth-limited search with depth limit 2, and iterative deepening depth-first search where in every iteration depth will be increased by 2. [*In-course. Marks: 8*]**
3.8	**What is state-space search? Contrast state-space search and dynamic programming. Can they be combined? [*2008, 2007. Marks: 3*]**

CHAPTER 4
INFORMED SEARCH & EXPLORATION

Short Notes

Target

- Chapter 3 showed that uninformed search strategies can find solutions to problems by systematically generating new states and testing them against the goal.
- Unfortunately, these strategies are incredibly inefficient in most cases.
- This chapter shows how an informed search strategy – one that uses problem-specific knowledge beyond the definition of the problem itself – can find solutions more efficiently.

Informed (Heuristic) Search Strategies

- The general approach we will consider is called *best-first search*.
- Best-first search is an instance of the general TREE-SEARCH or GRAPH-SEARCH algorithm in which a node is selected for expansion based on an evaluation function, $f(n)$.
- Traditionally, the node with the lowest evaluation is selected for expansion, because the evaluation measures distance to the goal.
- Best-first search can be implemented within our general search framework via a priority queue, a data structure that will maintain the fringe in ascending order of f-values.
- There is a whole family of BEST-FIRST-SEARCH algorithms with different evaluation functions.
- A key component of these algorithms is a heuristic function, denoted $h(n)$:

$$h(n) = \text{estimated cost of the cheapest path from node } n \text{ to a goal node.}$$

Greedy Best-First Search

- Greedy best-first search tries to expand the node that is closest to the goal, on the grounds that this is likely to lead to a solution quickly.
- Thus, it evaluates nodes by using just the heuristic function: $f(n) = h(n)$.
- As an example, we can use a *straight-line distance* heuristic – e.g. if the goal is Bucharest, we will need to know the straight-line distances to Bucharest – and find a path from Arad to Bucharest. See the work-through example on next page.
- **Optimal?** No. The path via Sibiu and Fagaras to Bucharest is 32 kilometers longer than the path through Rimnicu Vilcea and Pitesti.
- Resembles DFS in the way it prefers to follow a single path all the way to the goal, but will back up when it hits a dead end.
- **Disadvantages:** It suffers from the same defects as DFS:
    - It is not optimal, and it is incomplete (because it can start down an infinite path and never return to try other possibilities).
    - The worst-case time and space complexity is $O(b^m)$, where m is the maximum depth of the search space.

A* Search: Minimizing the Total Estimated Solution Cost

- The most widely-known form of best-first search.
- Evaluates nodes by combining $g(n)$, the cost to reach the node, and $h(n)$, the cost to get from the node to the goal:

$$f(n) = g(n) + h(n)$$

A Work-Through Example of Best-First Search Algorithm

Goal: Find a path from Arad to Bucharest using the *straight-line distance* heuristic

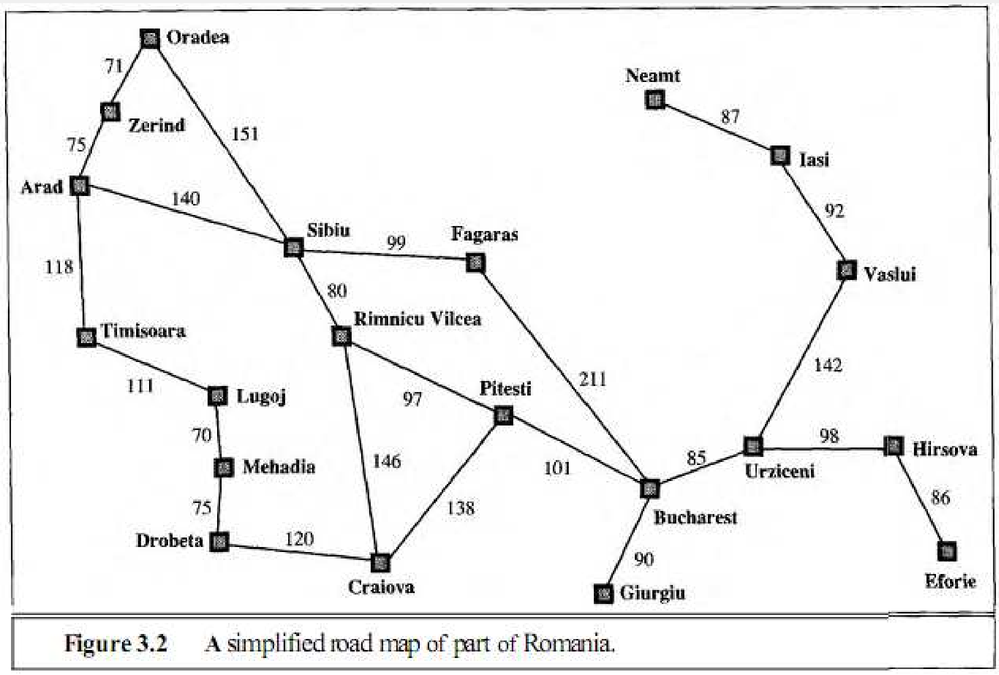
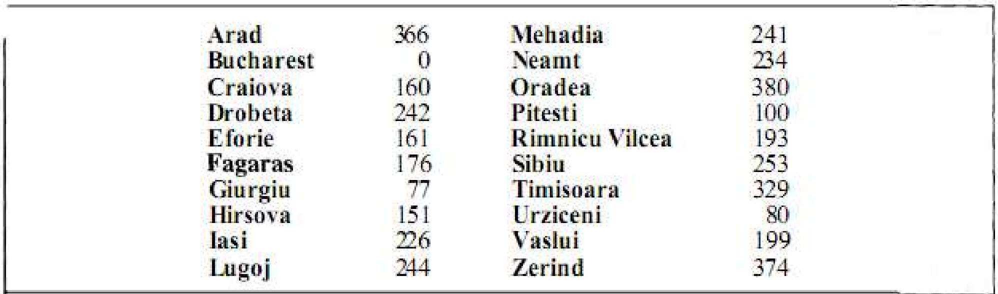
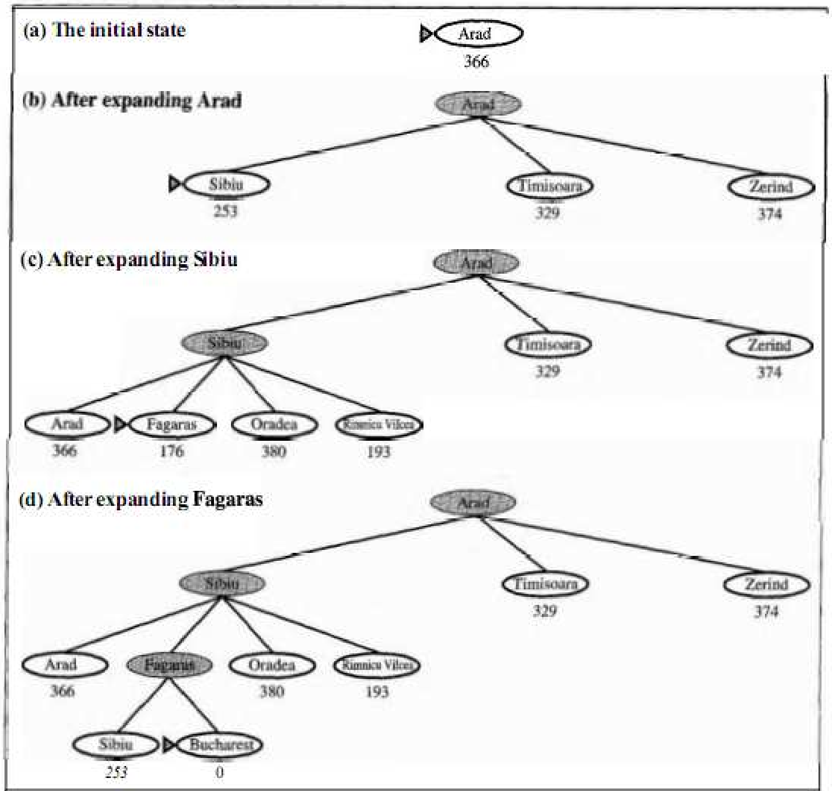


Figure 3.2 A simplified road map of part of Romania.



Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Figure 4.1 Values of h_{SLD}—straight-line distances to Bucharest.



(a) The initial state

(b) After expanding Arad

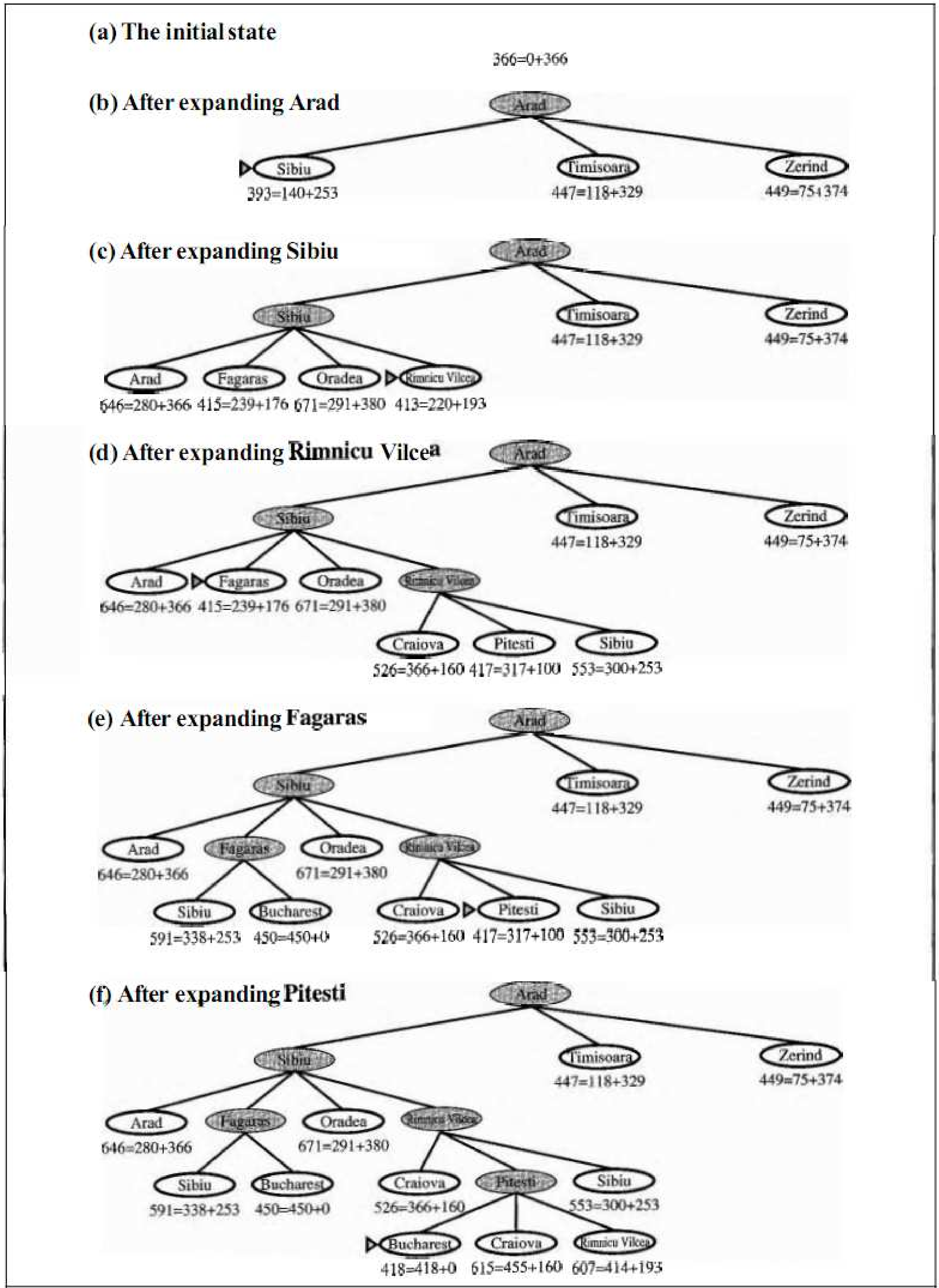
(c) After expanding Sibiu

(d) After expanding Fagaras

Figure 4.2 Stages in a greedy best-first search for Bucharest using the straight-line distance heuristic h_{SLD}. Nodes are labeled with their h-values.

A Work-Through Example of A* Search Algorithm

Goal: Find a path from Arad to Bucharest using the *straight-line distance* heuristic



(a) The initial state

(b) After expanding Arad

(c) After expanding Sibiu

(d) After expanding Rimnicu Vilcea

(e) After expanding Fagaras

(f) After expanding Pitesti

Figure 4.3 Stages in an A* search for Bucharest. Nodes are labeled with f = g + h. The h values are the straight-line distances to Bucharest taken from Figure 4.1.

- Since $g(n)$ gives the path cost from the start node to node n, and $h(n)$ is the estimated cost of the cheapest path from n to the goal, we have

  $$f(n) = \text{estimated cost of the cheapest solution through } n$$

  Thus, if we are trying to find the cheapest solution, a reasonable thing to try first is the node with the lowest value of $g(n) + h(n)$**.**

- **Optimal?**
  - A* using TREE-SEARCH is optimal if $h(n)$ is *admissible*, that is, provided that $h(n)$ never overestimates the cost to reach the goal.
    - Admissible heuristics are by nature optimistic, because they think the cost of solving the problem is less than it actually is.
    - Since $g(n)$ is the exact cost to reach n, we have as immediate consequence that $f(n)$ never overestimates the true cost of a solution through n.
  - A* using GRAPH-SEARCH is optimal if $h(n)$ is *consistent*.
    - A heuristic $h(n)$ is consistent if, for every node n and every successor n' of n generated by any action a, the estimated cost of reaching the goal from n is no greater than the step cost of getting to n' plus the estimated cost of reaching the goal from n':

      $$h(n) \leq c(n, a, n') + h(n')$$

    - Another important consequence of consistency is the following: *If h(n) is consistent, then the values of f(n) along any path are nondecreasing.*
      - From this, we can say that A* search is complete – as we add bands of increasing *f*, we must eventually reach a band where *f* is equal to the cost of the path to a goal state. However, completeness requires that there be only finitely many nodes with cost less than or equal to C*, a condition that is true if all step costs exceed some finite $\in$ and if *b* is finite.

- **Drawbacks:**
  - For most problems, the number of nodes within the goal contour search space is still exponential in the length of the solution.
    - For this reason, it is often impractical to insist on finding an optimal solution. One can use variants of A* that find suboptimal solutions quickly, or one can sometimes design heuristics that are more accurate, but not strictly admissible. In any case, the use of a good heuristic still provides enormous savings compared to the use of an uninformed search.
  - Because it keeps all generated nodes in memory, A* usually runs out of space long before it runs out of time.
    - For this reason, A* is not practical for many large-scale problems.
    - Recently developed algorithms (IDA*, RBFS, MA*, SMA*) have overcome the space problem without sacrificing optimality or completeness, at a small cost in execution time.

Memory-Bounded Heuristic Searches

1. Iterative-Deepening A* (IDA*)
2. Recursive Best-First Search (RBFS)
3. Memory-Bounded A* (MA*)
4. Simplified MA* (SMA*)

IDA*

- The simplest way to reduce memory requirements for A* is to adapt the idea of iterative deepening to the heuristic search context, resulting in the iterative-deepening A* (IDA*) algorithm.

- The main difference between IDA* and standard iterative deepening is that the cutoff used is the *f*-cost (*g* + *h*) rather than the depth; at each iteration, the cutoff value is the smallest *f*-cost of any node that exceeded the cutoff on the previous iteration.

Recursive Best-First Search (RBFS)

- A simple recursive algorithm that attempts to mimic the operation of standard best-first search, but using only linear space.
- Its structure is similar to that of a recursive depth-first search, but rather than continuing indefinitely down the current path, it keeps track of the f-value of the best alternative path available from any ancestor of the current node. If the current node exceeds this limit, the recursion unwinds back to the alternative path. As the recursion unwinds, RBFS replaces the *f*-value of each node along the path with the best *f*-value of its children. In this way, RBFS remembers the *f*-value of the best leaf in the forgotten sub-tree and can therefore decide whether it's worth re-expanding the sub-tree at some later time.

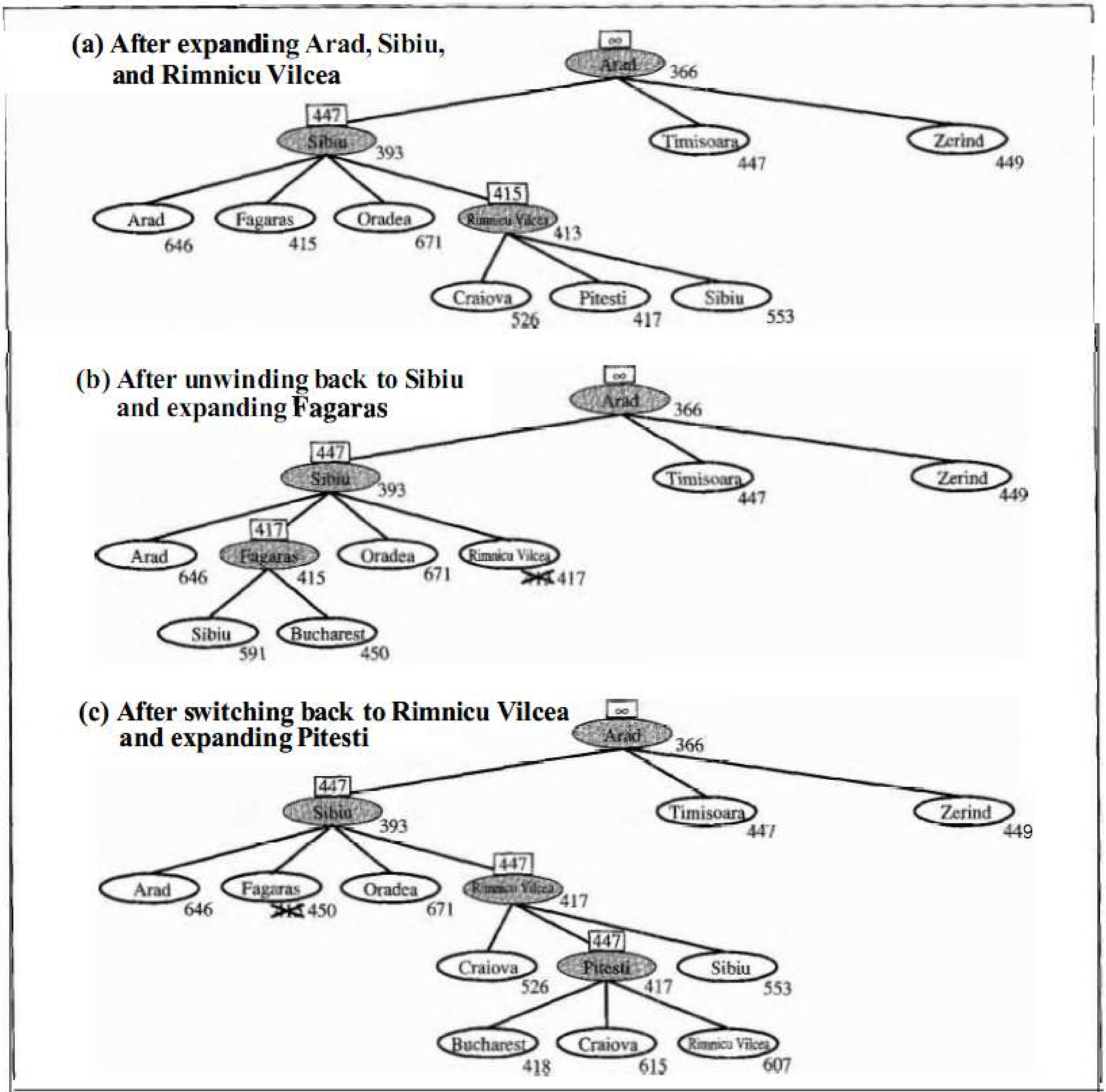


Figure 4.6 Stages in an RBFS search for the shortest route to Bucharest. The *f*-limit valuefor each recursive call is shown on top of each current node. (a) The path via Rimnicu Vilcea is followed until the current best leaf (Pitesti) has a value that is worse than the best alternativepath (Fagaras). (b) The recursion unwinds and the best leaf value of the forgotten subtree (417) is backed up to Rimnicu Vilcea; then Fagaras is expanded, revealing a best leaf value of 450. (c) The recursion unwinds and the best leaf value of the forgotten subtree (450) is backed up to Fagaras; then Rirnnicu Vilcea is expanded. This time, because the best alternativepath (through Timisoara)costs at least 447, the expansioncontinuesto Bucharest.

- **Advantage:** Somewhat more efficient than IDA*.
- **Disadvantage:** Still suffers from excessive node re-generation.
- **Optimal?** Yes, if the heuristic function *h*(*n*) is admissible.
- **Space complexity** is linear in the depth of the deepest optimal solution.
- **Time complexity** is rather difficult to characterize: it depends both on the accuracy of the heuristic function and on how often the best path changes as nodes are expanded.

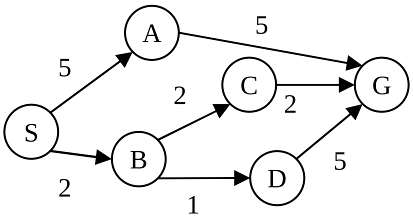
Disadvantage of Both IDA* and RBFS

- Both suffer from using too little memory. Between iterations, IDA* retains only a single number: the current *f*-cost limit. RBFS retains more information in memory, but it uses only linear space: even if more memory were available, RBFS has no way to make use of it.
- It seems sensible, therefore, to use all available memory. Two algorithms that do this are MA* (memory-bounded A*) and SMA* (simplified MA*).

4.1	**What is admissible heuristic? When does admissible heuristic become consistent? [*2003. Marks: 3*]**
4.2	**Describe the key dimensions that are necessary to analyze to implement any heuristic search algorithm to any problem. [*2003. Marks: 3*]**
4.3	**What is A* search and what is admissibility? Why is it important? What would happen if we used an inadmissible heuristic? [*2007. Marks: 3*]**
4.4	**In A* search, why does the heuristic have to always underestimate? [*2008, 2007. Marks: 2*]**
4.5	**What is the condition for A* to be optimal? Give an example when A* is not practical. [*2004. Marks: 4*]**
4.6	**A* is a best-first search for f(n) = g(n) + h(n) where g(n) is the cost to reach the node *n* and h(n) is a heuristic function from the node *n* to the goal. Now, given that the graph is a tree, choose special functions g(n) and h(n), as general as possible, that will make A* search become** **a. Breadth-first search** **b. Depth-first search** **c. Uniform-cost search [*In-course 1, 2008-2009. Marks: 6*]**
4.7	**Prove that uniform-cost search is a special case of A* search. [*In-course. Marks: 4*]**
4.8	**What do you understand by iterative deepening search? How can it be combined with A* search? [*2008. Marks: 3*]**
4.9	**Write whether each of the following statements is true or false: [*In-course 1, 2008-2009. Marks: 3*]** **a. Uniform-cost search is a special case of breadth-first search.** **b. Breadth-first search, depth-first search and uniform-cost search are special cases of best-first search.** **c. A* is a special case of uniform-cost search.**
4.10	**The major four criteria for evaluating search methods are: time complexity, space complexity, optimality and completeness. Using one or more of these criteria, attempt to justify the following statements: [*In-course 1, 2008-2009. Marks: 3*]** **a. Iterative deepening search is preferred over breadth-first search.** **b. Bidirectional search is preferred over breadth-first search.**

c. The A* algorithm is preferred over the hill-climbing method.

4.11 **Consider the following search space where we want to find a path from the start state S to the goal state G. The table shows three different heuristic functions h1, h2 and h3.**



Node	h1	h2	h3
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

a. **What solution path is found by Greedy Best-first search using h2? Break ties alphabetically.**

b. **What solution path is found by Uniform-Cost search? Break ties alphabetically.**

c. **Give the three found by algorithm A* using each of the three heuristic function, respectively. Break ties alphabetically. [*In-course 1, 2008-2009. Marks: 3 + 3 + 1 × 3*]**

4.12 **For each statement about Hill Climbing below, decide whether it's true or false, and give a one-sentence justification. [*In-course 1, 2008-2009. Marks: 3*]**

i. **There can be more than one global optimum.**

ii. **It is possible that every state is a local optimum. (A local optimum is defined to be a state that is *no better* than its neighbors.)**

iii. **Hill climbing with random restarts is guaranteed to find the global optimum if it runs long enough on a finite state space.**

4.13 **Discuss and compare hill climbing and best-first search techniques. [*2007, 2006. Marks: 3*]**

4.14 **Discuss and compare hill climbing and simulated annealing search techniques. [*2008. Marks: 2*]**

4.15 **Explain how simple hill climbing and steepest ascent hill climbing algorithms move to the next state from the current state. [*2003. Marks: 4*]**

4.16 **Give an example of an admissible heuristic for the eight puzzle problem. [*2007. Marks: 3*]**

ALSO, Consider trying to solve the 8-puzzle using hill climbing. Can you find a heuristic function that makes it work? Make sure it works on the following example:

Start

1	**2**	**3**
8	**5**	**6**
4	**7**	

End

1	**2**	**3**
4	**5**	**6**
7	**8**	

4.17 **Fifteen puzzle is like eight puzzle except there are fifteen tiles instead of eight. What is the branching factor of the search tree for fifteen puzzle? [*2007. Marks: 3*]**

4.18 **Give an example of an admissible heuristic for the fifteen puzzle problem. [*2008. Marks: 3*]**

Chapter 5
Constraint-Satisfaction Problems

5.1	**What do you understand by CSPs? How is backtracking search used for CSPs? Explain with example. [*2006. Marks: 2 + 5*]**
5.2	**What is cryptarithmetic problem? Give an example. [*2006. Marks: 3*]**
5.3	**Trace the constraint satisfaction procedure for solving the following cryptarithmetic problem: [*2005. Marks: 7*]** $\begin{array}{r} \mathbf{CROSS} \\ \underline{\mathbf{+ROADS}} \\ \mathbf{DANGER} \end{array}$ **Constraints:** **1. No two different letters can be assigned the same digit.** **2. The sum of the digits must be as shown in the problem.**
5.4	**Solve the following cryptarithmetic puzzle: [*In-course 1, 2008-2009. Marks: 2*]** **EARTH + URANUS = SATURN**

Chapters 7, 8 & 9
Propositional & First-Order Logics

7.1	**Define with example: Logic, Proposition Logic (PL), First-Order Predicate Logic (FOPL). [*2008. Marks: 3*]**
7.2	**What do you mean by FOPL (First-Order Propositional Logic)? Briefly discuss syntax and semantics for FOPL. [*2007. Marks: 4*]**
7.3	**Briefly discuss all the properties of statements in terms of PL or FOPL. [*2008. Marks: 3*]**
7.4	**What is the difference between declarative and procedural knowledge? [*2006. Marks: 3*]** [*Probably 1st para of page 197*]
7.5	**Translate the following sentences into propositional logic: [*In-course 1, 2008-2009. Marks: 2*]** **a. It will either rain today or it will be dry today.** **b. You will not pass this course unless you study.**
7.6	**Suppose you are building a knowledge-based system to plan the seating at a dinner party. You want to use the system to prove statements of the form OK(P_1, P_2) (or ¬OK(P_1, P_2)) means it is (or it is not) OK for person P_1 to sit next to person P_2. Suppose you give the following axioms to the system:** **$(\forall x, y)\ \text{Dislikes}(x, y) \rightarrow \neg\text{OK}(x, y)$** **$(\forall x, y)\ \text{Male}(x) \land \text{Male}(y) \rightarrow \neg\text{OK}(x, y)$** **$(\forall x, y)\ \text{Female}(x) \land \text{Female}(y) \rightarrow \neg\text{OK}(x, y)$** **$(\forall x, y)\ \neg\text{Dislikes}(x, y) \land \text{Male}(x) \land \text{Female}(y) \rightarrow \text{OK}(x, y)$** **$(\forall x, y)\ \neg\text{Dislikes}(x, y) \land \text{Female}(x) \land \text{Male}(y) \rightarrow \text{OK}(x, y)$** **$\text{Male}(\text{John}) \land \text{Female}(\text{Susan}) \land \text{Male}(\text{David}) \land \text{Female}(\text{Jane})$** **$\text{Dislikes}(\text{Susan}, \text{Dave}) \land \text{Dislike}(\text{Jane}, \text{John})$** **i. Do the axioms entail OK(Susan, John)? Explain.** **ii. Do the axioms entail ¬OK(Susan, John)? Explain.** **iii. Is the axiom in question (i) under Closed World Assumption? [*2004. Marks: 6 + 6 + 1*]**
7.7	**Can statement Q be derived logically from the following knowledge base? How (use either forward or backward chaining)? [*2006. Marks: 3*]** **$P \rightarrow Q$** **$L \land M \rightarrow P$** **$B \land L \rightarrow M$** **$A \land P \rightarrow L$** **$A \land B \rightarrow L$** **A** **B**
7.8	**What is CNF? Convert $A \Leftrightarrow B \land P \rightarrow R$ to CNF. [*2006. Marks: 4*]**
7.9	**Convert the following first-order sentences into CNF: [*In-course 1, 2008-2009. Marks: 3*]** **$\exists x\ \forall y\ \forall z\ (\text{person}(x) \land ((\text{likes}(x, y) \land y \neq z) \rightarrow \neg\text{likes}(x, z)))$**
7.10	**What is Inference Process? Distinguish between forward chaining and backward chaining reasoning with example. [*2008. Marks: 5*]** **ALSO, Explain the differences between forward and backward chaining and under what conditions each would be best to use for a given set of problems. [*2005. Marks: 7*]**
7.11	**Represent the following fact in predicate logic:** ***"For all the persons there must have parents and God gives His mercy to him who loves his***

parents and the person who gets God's mercy will be rewarded in the long run."

Convert the predicate logic into clause form. Show the process of conversion in details with the appropriate rule. [*2003. Marks: 7*]

7.12 **Consider the following sentences:**

- **John is not a good student**
- **If a student is not good then he/she hates science courses since these are hard.**
- **If a student hates a course then he doesn't like it.**
- **All the courses in the Basket Cleaning Department are easy.**
- **CS121 is a science course.**
- **BC431 is a basket cleaning course.**

i. Convert these sentences into formulas in predicate logic.
ii. Now convert these formulas into clause form.
iii. Use resolution to answer the question "What course would John like?" [*2005. Marks: 3 + 3 + 4*]

7.13 **Consider the following three sentences:**

- **There is a computer scientist who likes every operating system.**
- **Linux is an operating system.**
- **Someone likes Linux.**

i. Write a formula in first-order logic expressing each of the given facts. Call them A, B and C.
ii. Write the set of clauses corresponding to A, B and ¬C.
iii. Derive the empty clause from this set of clauses using resolution. [*In-course 1, 2008-2009. Marks: 9*]

7.14 **Consider the problem of finding clothes to wear in the morning. To solve this problem, it is necessary to use knowledge as:**

- **Wear jeans unless either they are dirty or you have a job interview today.**
- **Wear a sweater if it is cold.**
- **It is usually cold in the winter.**
- **Wear sandals if it is warm.**
- **It is usually warm in the summer.**

Show how the problem can be solved and how the solution changes as the relevant facts (such as time of year and dirtiness of jeans) change. [*2003. Marks: 4*]

7.15 **Show the steps required to put the following axiom/logic into clause form: [*2005.Marks: 3+3*]**

i. $\forall x : \forall y : [\text{Above}(x, y) \wedge \neg\text{On}(x, y)] \rightarrow \exists z : [\text{Above}(x, z) \wedge \text{Above}(z, y)]$
ii. $\forall x : [\text{Roman}(x) \wedge \text{Know}(x, \text{Marcus})] \rightarrow [\text{hate}(x, \text{Caesar}) \vee (\forall y : \exists z\ \text{hate}(y, z) \rightarrow \text{thinkcrazy}(x, y))]$

7.16 **Are the following two formulae valid? Substantiate your answer. [*2003. Marks: 6*]**

i. $\exists y\ \forall x\ p(x, y) \rightarrow \forall y\ \exists x\ p(x, y)$
ii. $\forall x\ \exists y\ p(x, y) \rightarrow \exists y\ \forall x\ p(x, y)$

7.17 **Find the most general Unifier for the following set of terms: [*2003. Marks: 5*]**

$$S = \{p(X, X, g(X), h(a)), p(f(Y), f(h(Z)), g(X), Y)\}$$

7.18 **What do you understand by Herbrand Universe and Herbrand Base? What are the conditions for an interpretation to be Herbrand? [*2003. Marks: 4*]**

7.19 **Consider the following logic program P:**

$p(a) \leftarrow p(x), q(x)$
$p(f(x)) \leftarrow p(x)$

	$q(b) \leftarrow$ $q(f(x)) \leftarrow q(x)$ **Write down the Herbrand Universe and Herbrand Base. Give the least Herbrand model. [*2003. Marks: 7*]**
7.20	**Consider the following set S of clauses:** $p \leftarrow r$ r $\leftarrow p, s$ $s \leftarrow p$ **Show that S is unsatisfiable by resolution. [*2003. Marks: 3*]**
7.21	**Consider the following set S of clauses:** $p \leftarrow q, r$ $q, r \leftarrow u, v$ u v **Is *p* a logical consequence of *S*? Is *~p* a logical consequence of *S*? Substantiate your answer. [*2003. Marks: 3*]**
7.22	**Find a model for the set of formula {p(a), ∃x ~p(x)}. [*2003. Marks: 4*]**

CHAPTER A
GENETIC ALGORITHMS

A.1	**What is genetic programming? Explain genetic algorithm with flowchart. [*2008. Marks: 6*]** **ALSO, What are the main steps of a genetic algorithm? Draw a flowchart that implements these steps. [*2006. Marks: 5*]** **ALSO, Discuss genetic algorithm. [*In-course 2, 2008-2009. Marks: 4*]**
A.2	**How is genetic algorithm applied in clustering? [*In-course 2, 2008-2009. Marks: 3*]**
A.3	**How does the crossover operator work? Give an example using fixed-length bit-strings. [*2006. Marks: 4*]**
A.4	**What are the differences between evolution and genetic algorithms? [*2006. Marks: 3*]**
A.5	**Explain the difference of the genetic operators – inversion, crossover and mutation. [*2005. Marks: 3*]**

CHAPTER B
NEURAL NETWORKS

B.1	**How does an artificial neural network model the brain? Distinguish between two major classes of learning paradigms: supervised learning and unsupervised learning. [*2007. Marks: 4*]** [*The first part is out of our syllabus. The second part is out of syllabus, too, as it is from topic 18.1. However, as our syllabus in general includes classes of learning, I've included this.* ☺]
B.2	**Write one advantage and one disadvantage of ANN as a machine learning model. [*In-course 2, 2008-2009. Marks: 2*]**
B.3	**What is learning in neural network? Give the learning algorithm for a single layer neural network. Explain the algorithm with suitable example. [*2003. Marks: 2 + 2 + 4*]**
B.4	**How do learning rate, threshold, number of iteration and number of errors relate the performance of a learning algorithm of a neural network? [*2008. Marks: 3*]**
B.5	**Explain the back-propagation network algorithm. [*2003. Marks: 5*]**
B.6	**How does a multi-layer neural network learn? Derive the back-propagation training algorithm. Demonstrate multi-layer network learning of the binary logic function Exclusive-OR. [*2007. Marks: 9*]** **ALSO, How does a Perceptron learn? Derive the back-propagation algorithm for a neural network. Demonstrate how a neural network learns for the binary logic function OR. [*2008. Marks: 6*]** [*The last parts in both questions are probably out of syllabus*]
B.7	**State back-propagation learning algorithm. Explain the real problems with back-propagation network. [*2005. Marks: 5 + 2*]**
B.8	**In a neural network with *n* nodes in the output layer, give one formula of calculating the total error. Using gradient descent error minimization, write down the weight adjustment formulas for edges leaving the output layer and the last hidden layer nodes. [*In-course 2, 2008-2009. Marks: 1 + 3*]**
B.9	**What is a multi-layer neural network? What is a hidden layer for, and what does it hide? [*2008. Marks: 3*]** **ALSO, What is hidden layer and what does it hide? [*2007. Marks: 3*]**
B.10	**Why is differentiable activation function necessary for multilayer neural network? [*2003. Marks: 3*]**

CHAPTER C
CLASSIFICATION & CLUSTERING (DUNHAM)

C.1 **Discuss the general setup of the supervised machine learning problem. [*In-course 2, 2008-2009. Marks: 2*]**

C.2 **What do you understand by the terms *entropy* and *information gain*? What is the role of information gain while determining the best split in a decision tree using the ID3 algorithm? [*In-course 2, 2008-2009. Marks: 3 + 1*]**

C.3 **Mention three differences between the ID3 and C4.5 techniques of building a decision tree. [*2008. Marks: 3*]**

C.4 **Write down two formulas to measure distances between two clusters of data points. [*In-course 2, 2008-2009. Marks: 2*]**

C.5 **The following table shows the mutual distances of five data points. Construct a dendogram using the single link technique. [*In-course 2, 2008-2009. Marks: 3*]**

	A	B	C	D	E
A	0	12	271	21	31
B		0	333	47	31
C			0	12	54
D				0	31
E					0

C.6 **Assuming three clusters, cluster the following data items using *k*-means clustering: {2, 4, 10, 12, 3, 20, 33, 11, 25}. [*In-course 2, 2008-2009. Marks: 3*]**

CHAPTER D
MISCELLANEOUS

D.1 **What is Bayesian reasoning (aka Bayes' rule / Bayes' theory)? [*2008, 2007, 2006. Marks: 1*]**

D.2 **How does an expert system rank potentially true hypotheses? Give an example. [*2006. Marks: 4*]**

[*Out of syllabus?? Or from chapter 20???* ☹]

D.3 **Why is the certainty factors theory considered as a practical alternative to Bayesian reasoning? [*2008, 2007, 2006. Marks: 3*]**

[*Probably out of syllabus??*]