



EEL5840: Elements of Machine Intelligence

Robobug

Announcements



- Reading Assignment:
 - > Nilsson chapter 9
- Announcements:
 - > Tentative 2nd Exam Dates:
 - 12/1/09 (Tuesday)
 - 12/3/09 (Thursday)
- Today's Handouts in WWW:
 - > Outline Class 20
- Web Site
 - > www.mil.ufl.edu/eel5840
 - > Software and Notes



University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Amato, Author

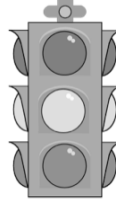
1

EEL5840: Elements of Machine Intelligence

Robobug

Today's Menu

- Finish class 19 slides {5-17}
- Heuristic Search (Chapter 9)
 - ⇒ The Use of Evaluation Functions
 - ⇒ A General Graph-Searching Algorithm
 - ⇒ Questions when you use a General Graph-Searching Algorithm



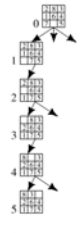
University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Amato, Author

2

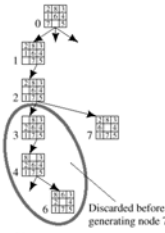
EEL5840: Elements of Machine Intelligence

Robobug

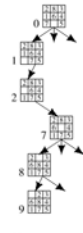
Search Strategies



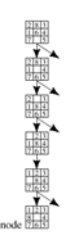
(a)



(b)



(c)



Goal node

Figure 8.3 Nodes generated with DFS, Bound=5

Figure 8.4
DFS Search

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Amato, Author

3

EEL5840: Elements of Machine Intelligence

Robobug

Search Strategies

PROCEDURE GRAPH-SEARCH

1. Create a *search graph*, G , consisting solely of the start node, s . Put s on a list called *OPEN*.
2. Create a list called *CLOSED* that is initially empty.
3. LOOP: if *OPEN* is empty, exit with failure.
4. Select the first node on *OPEN*, remove it from *OPEN*, and put it on *CLOSED*. Call this node n .
5. If n is a goal node, exit successfully with the solution obtained by tracing a path along the pointers from n to s in G . (see step 7.)
6. Expand node n , generating the set, M , of its successors and install them as successors of n in G .

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Amato, Author

4

EEL5840: Elements of Machine Intelligence

Robobug

Search Strategies

- Establish a pointer to n from those members of M that were not already in G (i.e., not already on either *OPEN* or *CLOSED*). Add these members of M to *OPEN*. For each member of M that was already on *OPEN* or *CLOSED*, decide whether or not to redirect its pointer to n . For each member of M already on *CLOSED*, decide for each of its descendants in G whether or not to redirect its pointer.
- Reorder the list *OPEN*, either according to some arbitrary scheme or according to heuristic merit.
- GO LOOP

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

5

EEL5840: Elements of Machine Intelligence

Robobug

Search Strategies

Node	Parent
s	-
R ₁₁	s
R ₁₂	R ₁₁
R ₂₁	R ₁₁
R ₂₂	R ₂₁
3	R ₂₁
R ₃₁	3
1	s
2	3
5	2
L ₁₁	s
L ₁₂	L ₁₁
L ₂₁	L ₁₁
6	L ₂₁
L ₆₁	6
4	6

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

6

EEL5840: Elements of Machine Intelligence

Robobug

Search Strategies

Node	Parent
s	-
R ₁₁	s
R ₁₂	R ₁₁
R ₂₁	R ₁₁
R ₂₂	R ₂₁
3	R ₂₁
R ₃₁	3
1	s
2	1
5	2
L ₁₁	s
L ₁₂	L ₁₁
L ₂₁	L ₁₁
6	L ₂₁
L ₆₁	6
4	2

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

7

EEL5840: Elements of Machine Intelligence

Robobug

Search Strategies

UNINFORMED SEARCH

- BREATH-FIRST SEARCH (BFS)**
 - > Reorder the nodes in step 8 so that the deepest nodes are considered last (shallowest nodes are considered first.)
 - > Analogous to $OPEN \leftarrow \text{append}(OPEN, M)$
- DEPTH-FIRST SEARCH (DFS)**
 - > Reorder the nodes on *OPEN* in step 8 so that the deeper nodes are considered first (shallowest nodes are considered last.)
 - > Nodes whose depth exceed a depth *BOUND* are never generated in step 6
 - > Analogous to $OPEN \leftarrow \text{append}(M, OPEN)$
 - > Also analogous to *BACKTRACKING*, except DFS generates *all* the successors in parallel, whereas *BACKTRACK* generates successors one at a time

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

8

EEL5840: Elements of Machine Intelligence

Robobug

Search Strategies

HEURISTIC GRAPH-SEARCH PROCEDURES

- The use of task-dependent information to help reduce search.
- It is often possible to specify heuristics that reduce search without sacrificing the guarantee of finding a minimal cost path length.
- We are usually interested in search methods that minimize the combination of costs of paths and cost of the search averaged over all problems likely to be encountered.
- If two methods solve a particular problem, the cheaper of the two is said to have more heuristic power.
- Heuristic information is used to order the nodes in step 8 of GRAPH-SEARCH by way of a real-valued evaluation function $f(n)$. By convention, nodes are ordered in increasing value of f .

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

9

EEL5840: Elements of Machine Intelligence

Robobug

Heuristic Search

- Using Evaluation Functions
 - > Suppose we order the nodes in step 8 of the Graph-Search Procedure preferentially via heuristic (problem-specific) information. Can we improve over LIFO or FIFO ordering?
 - > STEP 8 of Graph-Search is:
 - Reorder the list *OPEN*, either according to some arbitrary scheme or according to heuristic merit.
 - > We'll assume we have a heuristic evaluation function $f(n)$ to help us decide which node to expand next. $f(n)$ is a real-valued function over the domain of state descriptions.
 - > We expand next the node with smallest $f(n)$ resolving ties arbitrarily.
 - > We terminate when the node to be expanded next is a goal node

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

10

EEL5840: Elements of Machine Intelligence

Robobug

Search Strategies

Figure 9.1 Using $f'(n)=h'(n)$

Figure 9.2 Using $f'(n)=g'(n)+h'(n)$

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

11

EEL5840: Elements of Machine Intelligence

Robobug


Search Strategies

Example: Suppose you try the 8-Puzzle using the evaluation function $f(n)=d(n)+W(n)$ where $d(n)$ = depth(n) and $W(n)$ = number of misplaced tiles in DB_n

$f(s) = d(s) + W(s) = 0 + 4 = 4$
 $f(n_1) = d(n_1) + W(n_1) = 1 + 5 = 6$
 $f(n_2) = d(n_2) + W(n_2) = 1 + 3 = 4$
 $f(n_3) = d(n_3) + W(n_3) = 1 + 5 = 6$

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

12




EEL5840: Elements of Machine Intelligence

Heuristic Search

- QUESTIONS in Using Heuristic Evaluation Functions
 - > How do we settle on “good” evaluation functions?
 - > Are there any good general properties in Best-First search that can be exploited?
 - > Do we always find good paths to the goal? {Are there any guarantees?}
 - > What properties of $f(n)$ allow us to infer general properties about the search?

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

13



EEL5840: Elements of Machine Intelligence

The End!

University of Florida
EEL 5840 - Class #20 - Fall 2009
© Dr. A. Aravamudan

14