



EEL5840: Elements of Machine Intelligence

Robobug

## Announcements



- Announcements:
  - > Tentative 1<sup>st</sup> Exam Date  
Thur. Oct. 8<sup>th</sup> in class
- Today's Handouts in WWW:
  - > Outline Class 15
- Web Site
  - > [www.mil.ufl.edu/eel5840](http://www.mil.ufl.edu/eel5840)
  - > Software and Notes
  - > XLISP Documentation



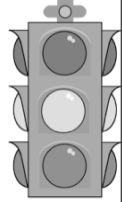
University of Florida  
EEL 5840 - Class #15 - Fall 2009  
© Dr. A. Amato Amato

EEL5840: Elements of Machine Intelligence

Robobug

## Today's Menu

- Machine Evolution
  - > Evolutionary Computation
  - > Genetic Programming
    - Program Representation in GP
    - The GP Process
    - Evolving a Wall-Following Robot



University of Florida  
EEL 5840 - Class #15 - Fall 2009  
© Dr. A. Amato Amato

EEL5840: Elements of Machine Intelligence

Robobug

## Machine Evolution

- Evolutionary Computation
  - > Is it possible to use the ideas in GAs to produce useful programs?
  - > Two uses of evolutionary processes: optimization, that is the process of attempting to find a maximum (minimum) in a given fitness function.
  - > To evolve programs to solve specific problems—for example, programs to control reactive robotic agents. The technique called genetic programming or *GP* for short is discussed in Nilsson's Chapter 4.

University of Florida  
EEL 5840 - Class #15 - Fall 2009  
© Dr. A. Amato Amato

EEL5840: Elements of Machine Intelligence

Robobug

## Machine Evolution

- Genetic Programming (*GP*)

The idea is to evolve functional programs, programs where all modules consist of functions with explicit return values (even if the returned value is void), e.g., LISP programs. Functional programs (*FPS*) can usually be expressed as rooted trees with labeled nodes.

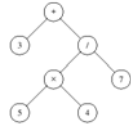


Figure 4.1  
A Program Tree

XLISP-PLUS version 3.04  
Portions Copyright (c) 1988, by David Betz.  
Modified by Thomas Almy and others.  
XLISP-STAT Release 3.52.16 (Beta).  
Copyright (c) 1989-1999, by Luke Tierney.

```
> (+ 3 (/ (* 5 4) 7))  
5.857142857142858  
>
```

University of Florida  
EEL 5840 - Class #15 - Fall 2009  
© Dr. A. Amato Amato



EEL5840: Elements of Machine Intelligence

Robobug

## Machine Evolution

- **GP Process**
  - > Start with a population of random programs to create generation 0
  - > Nilsson will start with 5,000 random programs constructed from the primitives AND, OR, NOT and IF, the actions {up, down, left, right}, the sensory inputs {n, ne, e, se, s, sw, w, nw} and the Boolean constants 0 and 1.
  - > Program evaluation is accomplished by running the program 60 times and counting the number of cells next to the wall that are visited during the 60 steps. Since there are 32 cells that constitute the wall, the maximum attainable by any one program execution is 32. Perform 10 such runs per trial starting in arbitrarily chosen (random) starting positions. Let  $f$  (the fitness function) be the total number of cells visited in these ten runs. The maximum of 320 would be obtained by a robot that achieves wall following.

University of Florida  
EEL 5840 - Class #13 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

## Machine Evolution

- **GP Process - Reproduction of the  $i+1$  generation from the  $i^{\text{th}}$** 
  - > Five hundred programs (10%) from generation  $i$  are copied directly into generation  $i+1$ . Individuals are chosen using a tournament selection process: seven programs are randomly selected (with replacement) from the population of 5,000. Then the most fit of these seven programs is chosen. (These values are determined empirically and by experience)
  - > 4,500 new child programs (90%) are put into generation  $i+1$ . Each child program is produced from a mother and a father program by a crossover operation as follows: a mother and a father are each selected using tournament selection as before. Then a randomly chosen subtree from the father replaces a randomly selected subtree from the mother resulting in the child.

University of Florida  
EEL 5840 - Class #13 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

## Machine Evolution

- **GP Process - Reproduction of the  $i+1$  generation from the  $i^{\text{th}}$**

University of Florida  
EEL 5840 - Class #13 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

## Machine Evolution

- **GP Process - Mutation**
  - > Sometimes a mutation operator is used in constructing individuals for the next generation. When used, it is used sparingly (perhaps at 1% rate or lower).
  - > The mutation operator selects a single parent from generation  $i$  by tournament selection. A randomly selected subtree is deleted from this parent and replaced by a newly grown random subtree created in the same manner as for the members of generation 0.
  - > Nilsson's example does not implement mutation.

University of Florida  
EEL 5840 - Class #13 - Fall 2009  
© Dr. A. Aravamudan



