Modern Robots:
Evolutionary Robotics

Jeff Clune
Assistant Professor
Evolving Artificial Intelligence Laboratory

UNIVERSITY OF WYOMING
Transferability Approach

Mouret, Koos, & Doncieux
Transferability Approach

• If you can’t be with the simulator you want, love the one you’re with

• aka: don’t fix the simulator, just recognize where it sucks

• Imagine:
  • in simulator, fastest way around track is to drift turns
    - simulated evolution drifts perfectly and is fast
  • in reality, drifting is unreliable
    - leads to lots of crashes = slow
  • how would you fix the problem (algorithmically)?
Transferability Approach

• OK, works when you know a behavior doesn’t transfer
  • but how can you learn what types of behaviors don’t transfer?
    - e.g. that skidding also doesn’t transfer well?
Transferability Approach

- Evolution has no reason not to exploit poorly modeled behaviors
  - e.g. cartwheeling crazily

- Humans tend to know not to trust certain parts of the model
  - e.g. rolling a car over a barricade to skip half a lap
  - can you think of other examples?

- But you can’t exhaustively test every type of behavior to see if it transfers poorly
Transferability Approach

• Robots
  • Bioloid

• Constrained encoding
  • each motor follows a sine wave
  • all sine waves are a function of $p_1$, $p_2$
  • allows mapping/visualizing entire space!
  - 5500 real robot experiments!!

\[ \alpha(i, t) = \frac{5\pi}{12} \cdot \text{dir}(i) \cdot p_1 - \frac{5\pi}{12} \cdot p_2 \cdot \sin(2\pi t - \phi(i)) \]

where $\alpha$ denotes the desired angular position of the motor $i$ at time-step $t$. $\text{dir}(i)$ is equals to 1 for both motors of the front-right leg and for both motors of the rear-left leg; $\text{dir}(i) = -1$ otherwise (see Fig. 1 for orientation). The phase angle $\phi(i)$ is 0 for the upper leg motors of each leg and $\pi/2$ for the lower leg motors of each leg. Both motors of one leg consequently have the same control signal with different phases. Angular positions of the actuators are constrained in
• First ever complete map of reality vs. simulation
• Very different (4 peaks in sim, 1 or 2 in reality, major valleys in sim)
• Only one peak in sim transfers (e.g. getting stuck in top left no good)
• But there is one good area, so finding it is the goal
Transferability Approach

• Claim:
  • most simulators can get some things right,
  • but can’t be perfect (and are often far from it)

• So, just learn which simulator parts not to use

• I.e. Learn the “Transferability Function”
Transferability Approach

• Transferability Function
  • transferability = f_t(phenotype)
    - Note: not fitness, just whether simulator and reality correspond
    - phenotype can be genotype, phenotype, or behavior
      - genotype usually harder if mappings are complex (e.g. generative encodings), behavior easiest
Transferability Approach

• Their exhaustive mapping of the transferability function
  • Treatment1: similarity measure = distance (fitness)
  • Treatment2: similarity measure = euclidean distance between COM trajectory points
  • Both reveal areas that don’t transfer well (e.g. top left peak)

Figure 4: (left) Transferability function based on the difference of fitness values. (right) Transferability function based on the difference of trajectories. In both maps, the contour line denotes the zones for which the simulation leads to high fitness values (greater than 900mm).
Transferability Approach

- **Transferability Function**
  - Can’t map real problems
  - So we need to *learn* the TF
    - Can use supervised learning
    - Anything that maps features to a value
      - Neural networks
      - SVMs
        - Example: all behaviors that somersault won’t transfer well.
    - Or, simpler, assume nearby points tend to have similar transferability
      - Then learn distance (or similarity) function
Transferability Approach

• Terms
  • Transferability
    - TF: transferability = f(phenotype)
    - ApproxTF: best guess of transferability = f(phenotype)
  • Fitness
    - FF: fitness function in reality = f(phenotype)
    - ApproxFF: best guess of fitness function in reality = f(phenotype)
Transferability Approach

• Why learn AproxTF & not an approx. fitness function?
  • AproxTF: transferability = f(phenotype)
  • ApproxFit: fitnessInReality = f(phenotype)
Transferability Approach

• Why learn AproxTF & not an approx. fitness function?
  • AproxTF: transferability = f(phenotype)
  • AproxFit: fitness\text{InReality} = f(phenotype)

• AproxFit is hard
  • it IS a good simulator (or at least a decent simulator and a TF)
    - thus would require learning laws of articulated rigid body dynamics

• AproxTF is easier
  • it doesn’t specify all the different ways that transferability is low or high
    - it can key in on simple features: e.g. if COM > 2, transferability is low
  • it thus complements a simulator, instead of reinventing it
Transferability Approach

• Overall goal: Find
  • High fitness solutions (in simulation)
  • That transfer well

• Note: often the most fit in sim don’t transfer
  • (e.g. highly dynamic behaviors)
  • so there will be a tradeoff

• Multi-objective algos to the rescue

\[
\text{maximize} \left\{ \begin{array}{l}
\text{fitness}(x) \\
\text{approximated transferability}(x)
\end{array} \right. 
\]
Transferability Approach

• What supervised learning algo to use to learn ApproxTF?
  • They assume nearby points have similar transferability
    - And use “Inverse Distance Weighting”
Transferability Approach

• When do we transfer?
  - Do N up front, don’t add
  - Add online during run
    - Active Learning approach: transfer the most different (least crowded)
    - Many other options
Figure 5: Principle of the multi-objective optimization of both the fitness and the transferability. Individuals from the population are periodically transferred on the robot to improve the approximation of the transferability function.
Transferability Approach

• Same ideas apply to generalization
  • Reality is general performance
  • A few trials are the simulation
  • Learn which types of behaviors transfer/generalize well
  • Pinville et al. 2011

• Same ideas apply to damage recognition
  • we’ll talk about that next
Transferability Approach

• Does it work?

• TF (used to check transferability):
  • distance traveled
  • average height of COM
  • robot heading at end of experiment
Transferability Approach

- 10 real evals allowed
- Controls
  - Evolution in hardware
  - Evolution in sim then transfer
  - Evolution in sim, then transfer, 10 hardware evals to further learn
    - stochastic gradient descent
  - Approx. Fitness
    - Inverse Distance Method and Kriging
Transferability Approach

• High performance in sim, but not in reality w/o Trans. Approach

• Similar results with another problem