Simultaneous Learning and Reshaping of an Approximated Optimization Task

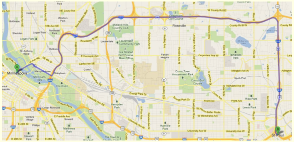
Patrick MacAlpine, Elad Liebman, and Peter Stone

Department of Computer Science, The University of Texas at Austin

May 6, 2013

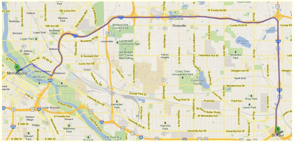
LARG
Learning Agents Research Group
The University of Texas at Austin

Motivation: A General Optimization Task

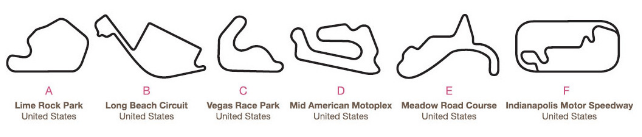


Goal: Optimize parameters for an autonomous vehicle for task of driving across town

Motivation: A General Optimization Task

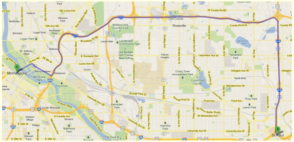


Goal: Optimize parameters for an autonomous vehicle for task of driving across town

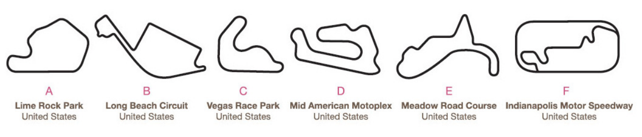


Optimization tasks: Different obstacle courses to drive car on

Motivation: A General Optimization Task



Goal: Optimize parameters for an autonomous vehicle for task of driving across town



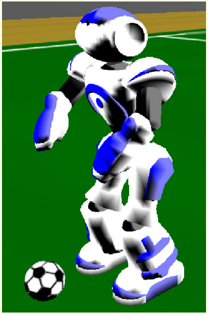
Optimization tasks: Different obstacle courses to drive car on

Research Question:

Which optimization task(s) to use for learning, and can we determine this while simultaneously optimizing parameters?

RoboCup 3D Simulation Domain

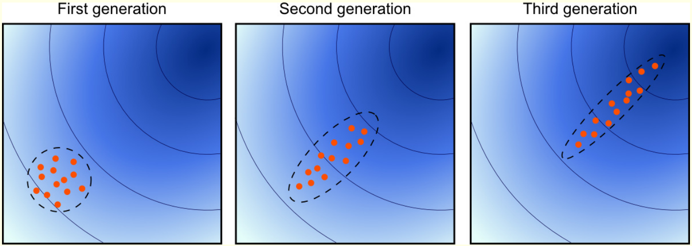
- Teams of 11 vs 11 autonomous robots play soccer
- Realistic physics using Open Dynamics Engine (ODE)
- Simulated robots modeled after Aldebaran Nao robot
- Robot receives noisy visual information about environment
- Robots can communicate with each other over limited bandwidth channel



Omnidirectional Walk Engine Parameters to Optimize

Notation	Description
$\text{maxStep}_{\{x,y,\theta\}}$	Maximum step sizes allowed for x, y, and θ
y_{shift}	Side to side shift amount with no side velocity
z_{torso}	Height of the torso from the ground
z_{step}	Maximum height of the foot from the ground
f_{g}	Fraction of a phase that the swing foot spends on the ground before lifting
f_{a}	Fraction that the swing foot spends in the air
f_{s}	Fraction before the swing foot starts moving
f_{m}	Fraction that the swing foot spends moving
ϕ_{length}	Duration of a single step
δ_{step}	Factor of how fast the step sizes change
x_{offset}	Constant offset between the torso and feet
x_{factor}	Factor of the step size applied to the forwards position of the torso
$\delta_{\text{target\{tilt,roll\}}}$	Factors of how fast tilt and roll adjusts occur for balance control
$\text{ankle}_{\text{offset}}$	Angle offset of the swing leg foot to prevent landing on toe
err_{norm}	Maximum COM error before the steps are slowed
err_{max}	Maximum COM error before all velocity reach 0
$\text{COM}_{\text{offset}}$	Default COM forward offset
$\delta_{\text{COM}\{x,y,\theta\}}$	Factors of how fast the COM changes x, y, and θ values for reactive balance control
$\delta_{\text{arm}\{x,y\}}$	Factors of how fast the arm x and y offsets change for balance control

CMA-ES (Covariance Matrix Adaptation Evolutionary Strategy)



(image from wikipedia)

- Evolutionary numerical optimization method
- Candidates sampled from multidimensional Gaussian and evaluated for their fitness
- Weighted average of members with highest fitness used to update mean of distribution
- Covariance update using evolution paths controls search step sizes

Obstacle Course Optimization Video

- Agent is measured on its cummulative performance across 11 activities
- Agent given reward for distance it is able to move toward active targets
- Agent is penalized it if falls over

Obstacle Course Optimization Activities

1. Long walks forward/backwards/left/right
2. Walk in a curve
3. Quick direction changes
4. Stop and go forward/backwards/left/right
5. Alternating moving left-to-right & right-to-left
6. Quick changes of target to simulate a noisy target
7. Weave back and forth at 45 degree angles
8. Extreme changes of direction to check for stability
9. Quick movements combined with stopping
10. Quick alternating between walking left and right
11. Spiral walk both clockwise and counter-clockwise

Evalution Function: 4v4 Game

- Teams of four agents play a 5 minute game against each other
- Team being evaluated plays against team using walk optimized with obstacle course

$$\text{Fitness}_{4v4} \quad = \quad \textit{goalsDifferential} * 15\{\frac{1}{2}\text{Field\_Length}\} + \textit{avgBallPosition}$$

Single Activity Analysis

$\text{Fitness}_{4v4} = 0$ in expectation for for optimizing across all 11 activities

Activity	Fitness_{4v4}	StdErr
1	-26.961	1.296
2	-31.250	1.088
3	-26.245	1.152
4	-23.779	1.074
5	-65.951	1.285
6	-66.005	0.912
7	-44.425	1.155
8	-79.694	0.941
9	-80.161	0.816
10	-68.743	0.958
11	-82.862	0.928

Single Activity Analysis

$\text{Fitness}_{4v4} = 0$ in expectation for for optimizing across all 11 activities

Activity	Fitness_{4v4}	StdErr
1	-26.961	1.296
2	-31.250	1.088
3	-26.245	1.152
4	-23.779	1.074
5	-65.951	1.285
6	-66.005	0.912
7	-44.425	1.155
8	-79.694	0.941
9	-80.161	0.816
10	-68.743	0.958
11	-82.862	0.928

No single activity gives as good or better performance than all activities combined.

Weighting Each Activity

Weights $= w_1 ... w_{11}$
Baseline $w_{i \in [1,11]} = 1$

Activity rewards $= r_1 ... r_{11}$

$$\text{reward} = \sum_{i \in [1,11]} w_i \cdot r_i$$

where r_i is the activity reward from the i-th activity and w_i is its weight

Want to learn weights that improve performance of fitness$_{4v4}$ simultaneously as we optimize parameters for the walk engine.

Weighting Each Activity

Weights $= w_1 ... w_{11}$
Baseline $w_{i \in [1,11]} = 1$

Activity rewards $= r_1 ... r_{11}$

$$\text{reward} = \sum_{i \in [1,11]} w_i \cdot r_i$$

where r_i is the activity reward from the i-th activity and w_i is its weight

Want to learn weights that improve performance of fitness$_{4v4}$ simultaneously as we optimize parameters for the walk engine...otherwise weighting problem becomes waiting problem.

Activity Weight Analysis

Activity	$\text{Fitness}_{4v4}, w_i = 0$	$\text{Fitness}_{4v4}, w_i = 2$
1	5.142	1.126
2	1.529	5.238
3	-23.076	-0.373
4	-12.437	4.720
5	0.181	-3.659
6	1.801	-1.321
7	-0.997	5.325
8	4.262	-6.358
9	-7.979	-3.077
10	2.473	-18.182
11	2.403	4.203

Colors represent statistically significant positive and negative fitness
All standard errors less than 1.76

Activity Weight Analysis

Activity	$\text{Fitness}_{4v4}, w_i = 0$	$\text{Fitness}_{4v4}, w_i = 2$
1	5.142	1.126
2	1.529	5.238
3	-23.076	-0.373
4	-12.437	4.720
5	0.181	-3.659
6	1.801	-1.321
7	-0.997	5.325
8	4.262	-6.358
9	-7.979	-3.077
10	2.473	-18.182
11	2.403	4.203

Colors represent statistically significant positive and negative fitness

All standard errors less than 1.76

Baseline combination of all equal weights of 1 is not optimal

Learning Weights

- Run 4v4 evaluation of population members every 10th generation of CMA-ES

- Compute least squares regression between activity rewards and the 4v4 evaluation task reward

  Find w vector such that
  reward $= \sum_{i \in [1,11]} w_i \cdot r_i \approx \text{fitness}_{4v4}$

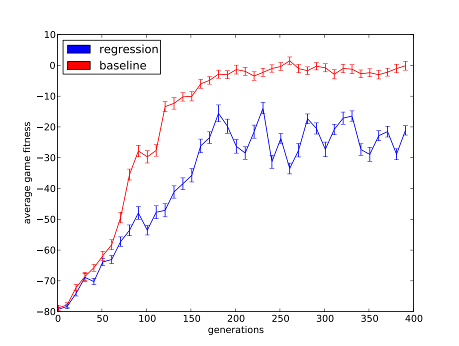
- Update weights for each activity based on the computed regression coefficients

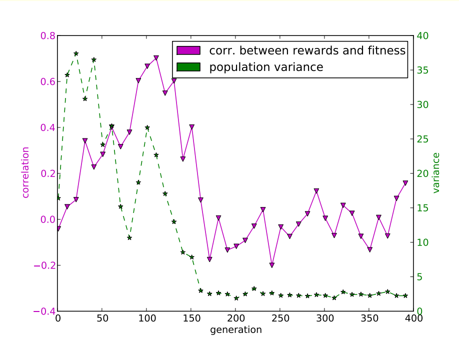
Negative Weights

- Allowing for negative weights is bad as it encourages poor performace on tasks
- Must use non-negative least squares regression or set negative weights equal to zero so as to not have negative weights

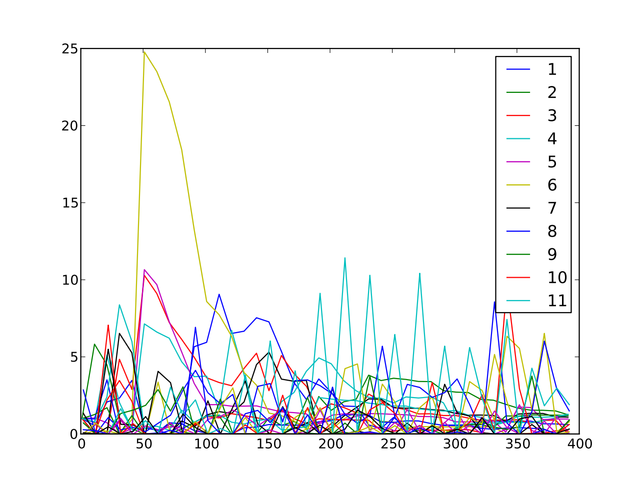
Population Convergence

- Correlation drops close to zero amplifying noise





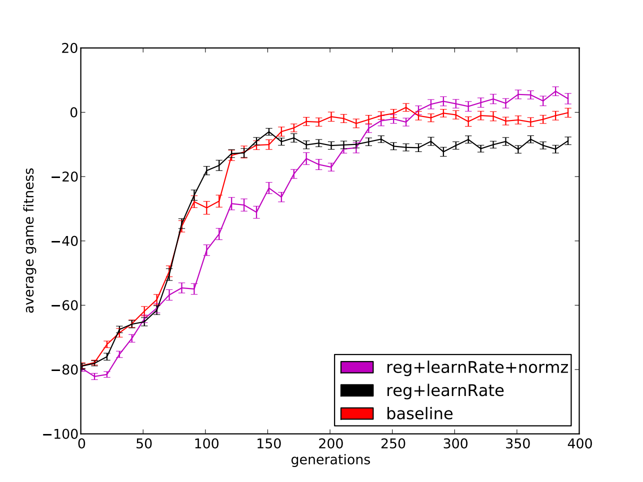
Regression Activity Weights



- Weights don't converge

Learning Rate and Normalization

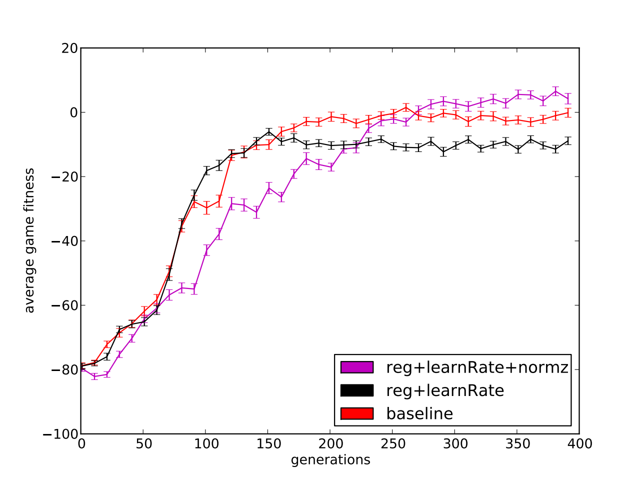
- Compute correlation of act. rewards to Fitness_{4v4} for learning rate $w_i = \text{lastWeight}_i + (\text{currentWeight}_i - \text{LastWeight}_i) * |\text{correlation}_i|$
- Use z-score based normalization for each activity reward such that $r_i = \frac{r_i - \bar{r}_i}{\sigma_i}$



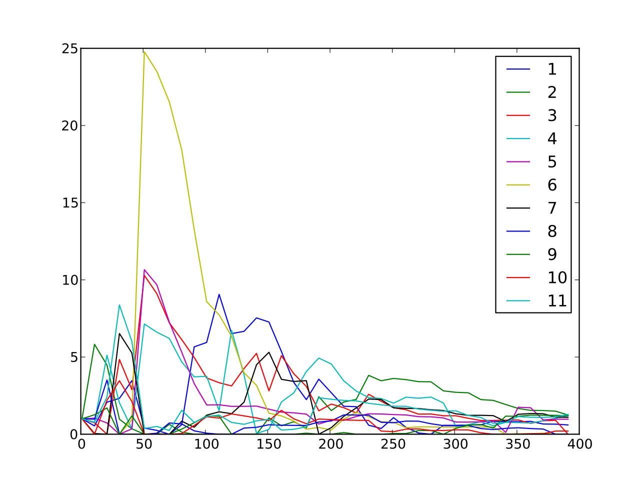
Learning Rate and Normalization

- Compute correlation of act. rewards to Fitness_{4v4} for learning rate $w_i = \text{lastWeight}_i + (\text{currentWeight}_i - \text{LastWeight}_i) * |\text{correlation}_i|$
- Use z-score based normalization for each activity reward such that $r_i = \frac{r_i - \bar{r}_i}{\sigma_i}$

Best value 6.535 (1.399)



Activity Weights



- Weights begin to converge
- Highest weight activities: spirals, stop and go, weave
- Zero weight activities: quick direction change, noisy target, extreme movements, quick alternating directions

Future Work

Watching 100s of simulated soccer games

Future Work

Watching 100s of simulated soccer games

Future Work

- Experiment with different activities for an obstacle course
  - Infant walk trajectories
  - Record walk trajectories from gameplay

Watching 100s of simulated soccer games

Future Work

- Experiment with different activities for an obstacle course
  - Infant walk trajectories
  - Record walk trajectories from gameplay

- Automate the construction of activities by learning/evolving activities during the course of optimization

Watching 100s of simulated soccer games

More Information

UT Austin Villa 3D Simulation Team homepage:
www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/

Email: patmac@cs.utexas.edu

More Information

UT Austin Villa 3D Simulation Team homepage:
www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/

Email: patmac@cs.utexas.edu

AUSTIN VILLA
ROBOT SOCCER TEAM
THE UNIVERSITY OF TEXAS AT AUSTIN

Wedesday at 12:20, Session A1 - Robotics I:
Humanoid Robots Learning to Walk Faster: From the Real World to Simulation and Back

Cummulative Approach

- Compute correlation across all generations
- Use z-score based normalization for each activity reward such that $r_i = \frac{r_i - \bar{r}_i}{\sigma_i}$