REINFORCEMENT LEARNING WITH COMPACT FUNCTION APPROXIMATION: A CASE STUDY FROM THE ELEVATOR DISPATCHING DOMAIN

ML-95 Workshop
Lake Tahoe
9 July 1995

Bob Crites
University of Massachusetts
crites@cs.umass.edu

A paper with more details is available from:
ftp://ftp.cs.umass.edu/pub/anw/pub/crites/nips8.ps.Z

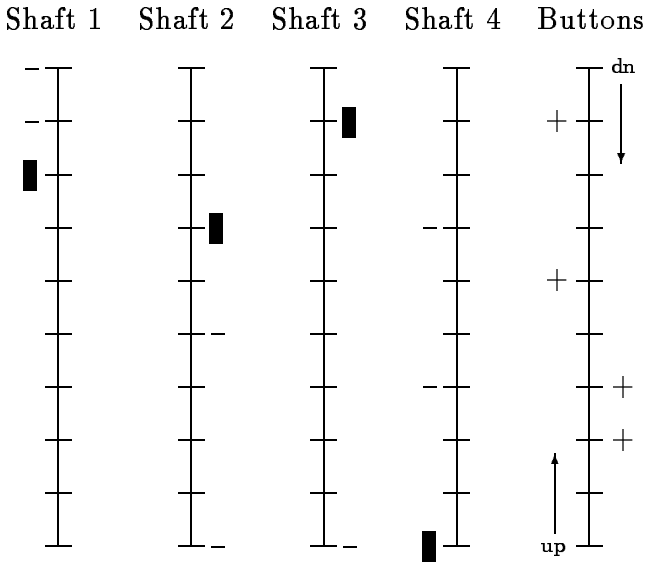
Acknowledgements: A. Barto, J. McNulty, C. Cassandras, A. Gandhi, D. Pepyne, V. Lesser, R. Grupen, R. Sutton, S. Bradtke, and the UMASS adaptive networks group. This research was supported by the Air Force Office of Scientific Research under grant F49620-93-1-0269.

OUTLINE OF TALK

- Elevator system
- Algorithm and network architecture
- Results
- Conclusions concerning value function approximation

ELEVATOR TASK:

Control 4 elevator cars in a simulated 10-story office building:



The elevators cars are represented as filled boxes in the diagram. '+' represents a *hall call* or someone wanting to *enter* a car. '−' represents a *car call* or someone wanting to *leave* a car. The left side of a shaft represents upward moving cars and calls. The right side of a shaft represents downward moving cars and calls.

ELEVATOR MODEL PARAMETERS

- Floor time (time to move one floor at max speed): 1.45 secs.
- Stop time (time to decelerate, open and close doors, and accelerate again): 7.19 secs.
- Turn time (time needed by a stopped car to change directions): 1 sec.
- Load time (time for one passenger to enter or exit a car): random variable with range from 0.6 to 6.0 secs, mean of 1 sec.
- Car capacity: 20 passengers.

PASSENGER ARRIVALS

- Non-stationary Poisson arrivals.
- Rates change every 5 minutes.
- Down-peak traffic profile developed by a major elevator company.

STATES

- Hall call buttons: 2^{18} combinations.
- Car call buttons: 2^{40} combinations.
- Positions and directions of cars: 18^4 combinations.
- Motion states of cars (accelerating, moving, decelerating, stopped, loading, turning)
- Set of passengers waiting at each floor, and each passenger's arrival time and destination: unobservable. However, 18 real numbers are available giving elapsed time since hall buttons were pushed.
- Set of passengers riding each car, and each passenger's destination: observable only through car calls.

CONSERVATIVE SIZE ESTIMATE of discrete approximation:

$$2^{18} \times 2^{40} \times 18^4 \approx 10^{22} \text{ states!}$$

ACTIONS

- When moving (halfway between floors): either stop at the next floor or continue past the next floor.
- When stopped at a floor: either go up or go down.
- Actions taken by the cars are asynchronous.

CONSTRAINTS:

- An elevator cannot pass a floor if a passenger wants to get off there.
- An elevator cannot change direction until it has serviced all passengers travelling in the current direction.
- Given a choice, always move up.
- Don't stop at a floor if another car is already stopping or stopped there.
- Don't stop at a floor unless someone wants to get on or off there.

PERFORMANCE CRITERIA

Minimize:

1. Average wait time.
2. Average squared wait time.
3. Average system time (wait plus travel time).
4. Percent waiting more than T seconds (e.g. where $T = 60$).

- Out of the four possibilities listed above, we used the squared wait as our criterion to encourage both fast and fair service.
- Instantaneous cost r equals the sum of the current squared waiting times of all waiting passengers.
- Define return as an integral rather than as an infinite sum:

$$\sum_{t=0}^{\infty} \gamma^t r_t$$

becomes

$$\int_0^{\infty} e^{-\beta\tau} r_\tau d\tau$$

Ref: S. J. Bradtke and M. O. Duff. (1995) Reinforcement Learning Methods for Continuous-Time Markov Decision Problems. In: G. Tesauro, D. S. Touretzky and T. K. Leen, eds., *Advances in Neural Information Processing Systems 7*, MIT Press, Cambridge MA.

ALGORITHM

- Team of continuous-time, discrete-event Q-learning agents.
- Parallel version uses one shared Q-function for all cars.
- Distributed version uses a separate Q-function for each car.
- Actions are selected using Boltzmann exploration with annealing.
- Q-values are stored in feedforward neural networks trained with error backpropagation.

$$\Delta\hat{Q}(x,a) = \alpha \cdot [\int_{t_1}^{t_2} e^{-\beta(\tau - t_1)} r_\tau d\tau + e^{-\beta(t_2 - t_1)} \min_b \hat{Q}(y,b) - \hat{Q}(x,a)],$$

NETWORK ARCHITECTURE FOR PURE DOWN TRAFFIC

- 47 input units: 18 for hall buttons, 16 for location and direction of car whose decision is required, 10 for location and direction of all other cars (superimposed), 1 highest queue bit, 1 longest queue bit, and 1 bias unit.
- 30 hidden sigmoid units.
- 1 or 2 linear output units (action value estimates). Little difference was found between separate and combined action networks.

PERFORMANCE COMPARISON

- SECTOR: sector-based algorithm.
- DLB: dynamic load balancing (attempts to equalize load on all cars).
- HUFF: highest unanswered floor first.
- LQF: longest queue first.
- FIM: finite intervisit minimization (receding horizon controller).
- ESA: empty the system algorithm (receding horizon controller).
- RLd: reinforcement learning, distributed.
- RLp: reinforcement learning, parallel.

Ref: G. Bao, C. G. Cassandras, T. E. Djaferis, A. D. Gandhi, D. P. Looze (1994) *Elevator Dispatchers for Down Peak Traffic.* ECE Dept Technical Report, University of Massachusetts, Amherst, MA.

PERFORMANCE COMPARISON: DOWN TRAFFIC ONLY

Algorithm	AvgWait	SquaredWait	SystemTime	Percent>60 secs
SECTOR	21.4	674	47.7	1.12
DLB	19.4	658	53.2	2.74
BASIC HUFF	19.9	580	47.2	0.76
LQF	19.1	534	46.6	0.89
HUFF	16.8	396	48.6	0.16
FIM	16.0	359	47.9	0.11
ESA/nq	15.8	358	47.7	0.12
ESA	15.1	338	47.1	0.25
RLd	15.5	366	42.4	0.49
RLp	14.8	318	42.1	0.11

PERFORMANCE COMPARISON: UP AND DOWN TRAFFIC

Algorithm	AvgWait	SquaredWait	SystemTime	Percent>60 secs
SECTOR	27.3	1252	54.8	9.24
DLB	21.7	826	54.4	4.74
BASIC HUFF	22.0	756	51.1	3.46
LQF	21.9	732	50.7	2.87
HUFF	19.6	608	50.5	1.99
ESA	18.0	524	50.0	1.56
FIM	17.9	476	48.9	0.50
RLd	18.7	615	45.7	2.28
RLp	17.2	490	42.9	1.56

PERFORMANCE COMPARISON: TWICE AS MUCH UP TRAFFIC

Algorithm	AvgWait	SquaredWait	SystemTime	Percent>60 secs
SECTOR	30.3	1643	59.5	13.50
HUFF	22.8	884	55.3	5.10
DLB	22.6	880	55.8	5.18
LQF	23.5	877	53.5	4.92
BASIC HUFF	23.2	875	54.7	4.94
FIM	20.8	685	53.4	3.10
ESA	20.1	667	52.3	3.12
RLd	20.5	737	49.0	3.87
RLp	19.3	640	46.3	3.04

CONCLUSIONS

- Another example of the power of reinforcement learning with compact function approximation.
- Importance of focusing computation on areas of state space most relevant to optimal control.
    1. The use of constraints improved the final level of performance.
    2. Training on heavy traffic was helpful because it increased the amount of experience in important areas of the state space.
    3. Training on actual trajectories can provide useful focusing.
- Weights *can* blow up. Especially when the learning rate is too high, the traffic profile is heavy (giving the network large input values), or the weights have just been randomly initialized.
- Training with pure down traffic and then including up traffic during testing worked better than training on the mixed traffic.
- Getting this to work has been difficult.