EXAMPLES

- CHESS PLAYER
  - MOVE INFORMED BY PLANNING + JUDGEMENTS OF DESIRABILITY OF PARTICULAR POSITIONS
- GAZELLE CALF
  - STRUGGLES TO FEET IN A FEW MINUTES
  - RUNNING AT 20 mph ½ HOUR LATER
- YOU PREPARING BREAKFAST
  - BEHAVIOR + INTERLOCKING GOAL/SUBGOAL RELATIONSHIPS

    WALK TO CUPBOARD, OPEN, SELECT, REACH, GRASP, RETRIEVE, ...

    ALSO: BOIL, STOOL, MILK, ...

APPLICATIONS [SUTTON & BARTO]

ARTHUR SAMUEL'S CHECKERS PLAYER

- FIRST PROGRAM, 1952
- LEARNING, 1956 (TV DEBUT 1956)
- GOOD, NOT EXPERT
- 2 IMPORTANT PAPERS (1959 + 1967)
- α-β SEARCH, "FEATURES"
- FOCUS ON GAMES FOR LEARNING (+ SHANNON, CHESS, 1950)

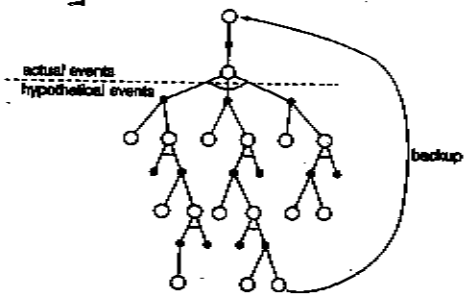
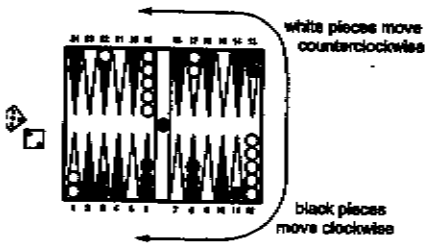


Figure 11.3 The backup diagram for Samuel's checkers player.



GERALD TESAURO'S TD-GAMMON (1992–1995)

Program	Training Games	Opponents	Results
TD-Gam 0.0	300,000	other programs	tied for best
TD-Gam 1.0	300,000	Robertie, Magriel, ...	−13 points / 51 games
TD-Gam 2.0	800,000	various Grandmasters	−7 points / 38 games
TD-Gam 2.1	1,500,000	Robertie	−1 point / 40 games
TD-Gam 3.0	1,500,000	Kazaros	+6 points / 20 games

MAXIMIZE: # OF POINTS (OR PROB. OF WINNING)

APPLICATIONS (CONT)

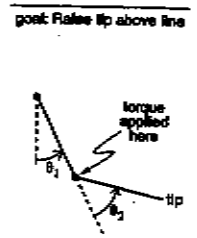
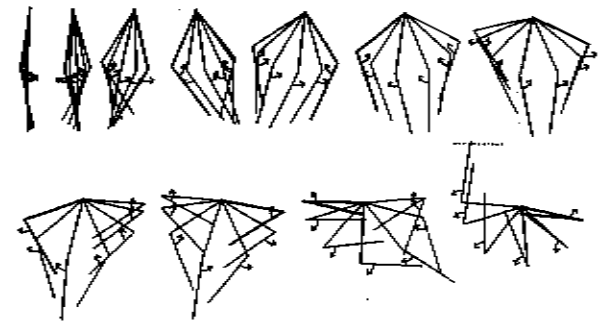


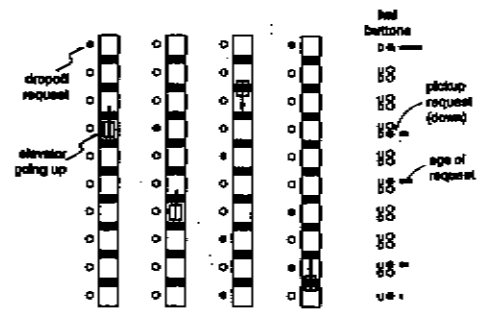
Figure 11.4 The acrobot.

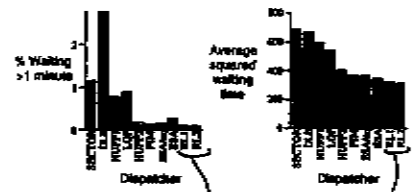


$$\ddot{\theta}_1 = -d_1^{-1}(d_2\ddot{\theta}_2 + \phi_1)$$

$$\ddot{\theta}_2 = \left(m_2 l_{c2}^2 + I_2 - \frac{d_2^2}{d_1}\right)^{-1}\left(\tau + \frac{d_2}{d_1}\phi_1 - m_2 l_1 l_{c2}\dot{\theta}_1^2 \sin\theta_2 - \phi_2\right)$$

ACROBOT (−1 REWARD UNTIL REACH GOAL)





REINFORCEMENT LEARNING (RL) (10^{22} STATES; REALISTIC SQ. OF REL TYPES)

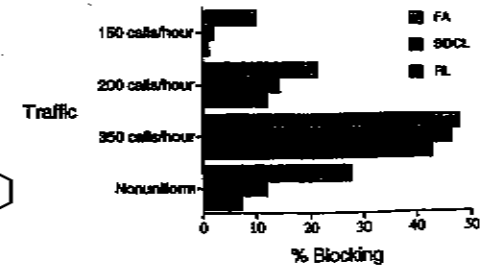
ELEVATOR DISPATCHING

DYNAMIC CHANNEL ALLOCATION

- ASSIGN CELL PHONE
- CAN'T REUSE IN ADJ. CELLS
- REWARDED FOR MORE CALLS

CAPACITY = 2 NEW CALL IS BLOCKED

CAPACITY = 3



ISSUES IN R-L

→ THORNDIKE'S LAW OF EFFECT:

Of several responses made to the same situation, those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation, so that, when it recurs, they will be more likely to recur; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections with that situation weakened, so that, when it recurs, they will be less likely to occur. The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond. (Thorndike, 1911, p. 244)

→ MINIMIZE/MAXIMIZE A PERFORMANCE MEASURE OF A DYNAMICAL SYSTEM OVER TIME

(OPTIMAL CONTROL, 1950's) E.G., ACROBOT, CLIMB TO ALTITUDE

→ COMBINES SEARCH + MEMORY

SEARCH (SELECTIONAL): TRY ALTERNATIVES, SELECT BY COMPARING CONSEQUENCES

e.g., NATURAL SELECTION

MEMORY (ASSOCIATIVE): - ALTERNATIVES ASSOC. W/ PARTICULAR SITUATION
- REMEMBER WHAT WORKED BEST + SITUATIONS IN WHICH WORKED BEST

e.g., SUPERVISED LEARNING/CLASSIFICATION

- STRENGTHEN ACTION IN STATE IF LEADS TO REWARD
- WEAKEN " " " " " " PUNISHMENT

→ TRIAL + ERROR SEARCH: NOT TOLD, MUST DISCOVER

→ CREDIT ASSIGNMENT PROBLEM (MINSKY, 1961)

DELAYED REWARD: ACTIONS AFFECT NEXT SITUATION, + THROUGH THAT, ALL SUBSEQUENT REWARDS

e.g., CHESS, ACROBOT

→ EXPLORATION/EXPLOITATION TRADEOFF:
- EXPLOIT WHAT KNOWS TO GET REWARD
- EXPLORE TO MAKE BETTER SELECTIONS IN FUTURE

{ E.G., FOSBURY FLOP 1968 GOLD 13/16 BY 1980 }

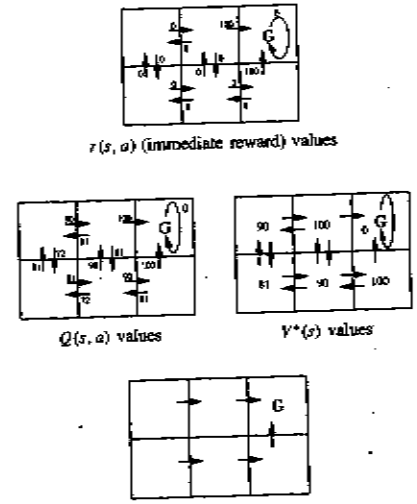
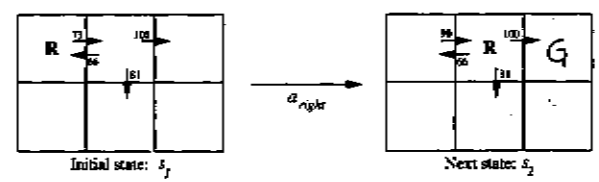


FIGURE 13.2
A simple deterministic world to illustrate the basic concepts of Q-learning. Each grid square represents a distinct state, each arrow a distinct action. The immediate reward function, $r(s,a)$ gives reward 100 for actions entering the goal state G, and zero otherwise. Values of $V^*(s)$ and $Q(s,a)$ follow from $r(s,a)$, and the discount factor $\gamma = 0.9$. An optimal policy, corresponding to actions with maximal Q values, is also shown.



$$\hat{Q}(s_1, a_{right}) \leftarrow r + \gamma \max_{a'} \hat{Q}(s_2, a')$$
$$\leftarrow 0 + 0.9 \max\{66, 81, 100\}$$
$$\leftarrow 90$$

FIGURE 13.3
The update to $\hat{Q}$ after executing a single action. The diagram on the left shows the initial state s_1 of the robot (R) and several relevant $\hat{Q}$ values in its initial hypothesis. For example, the value $\hat{Q}(s_1, a_{right}) = 72.9$, where a_{right} refers to the action that moves R to its right. When the robot executes the action a_{right}, it receives immediate reward $r = 0$ and transitions to state s_2. It then updates its estimate $\hat{Q}(s_1, a_{right})$ based on its $\hat{Q}$ estimates for the new state s_2. Here $\gamma = 0.9$.

[SOURCE: MITCHELL, *Machine Learning*]