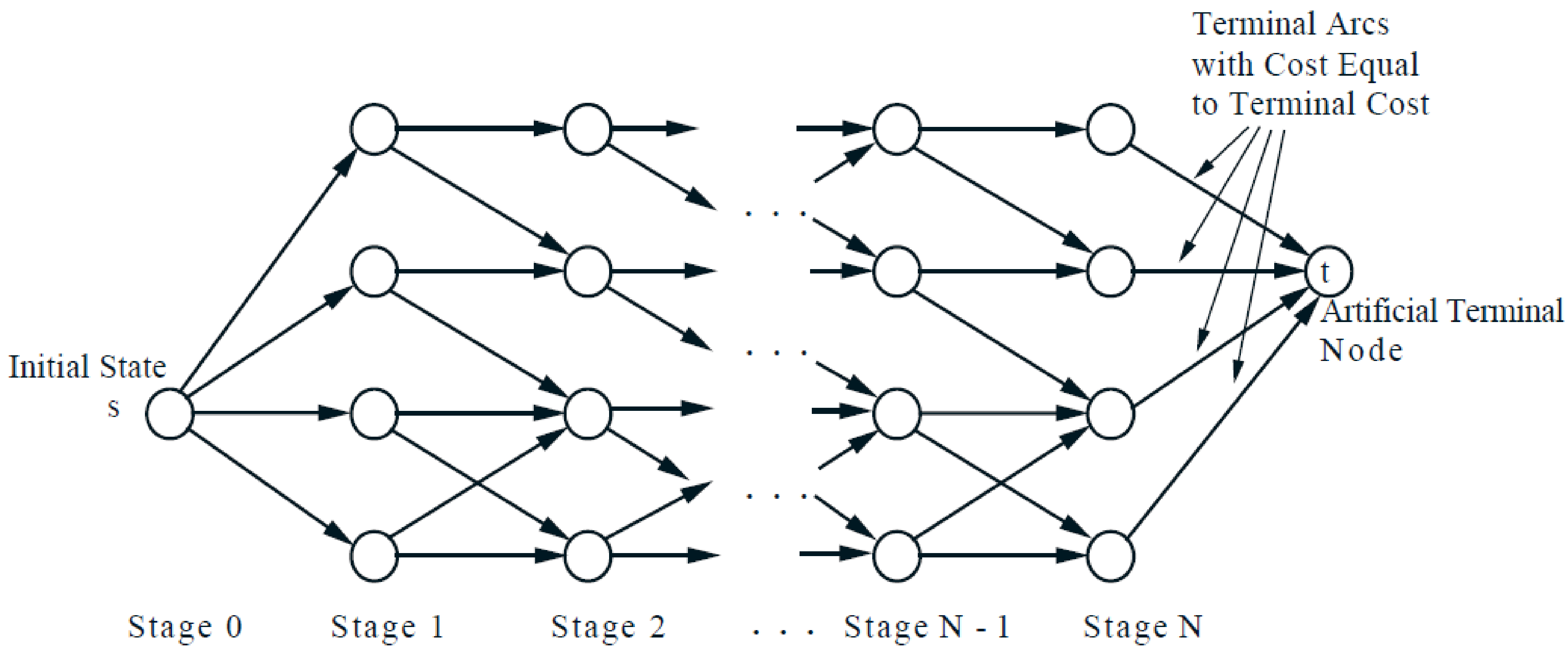
动态规划问题举例
Examples in DP

电信学院·自动化科学与技术系
系统工程研究所
吴江

Outline

- 确定性定期多阶段决策问题
- 确定性不定期多阶段决策问题

状态转移图


Terminal Arcs
with Cost Equal
to Terminal Cost
Initial State
s
t
Artificial Terminal
Node
Stage 0
Stage 1
Stage 2
. . .
Stage N - 1
Stage N

基本递推方程

$$\boxed{f_k(x_k) = \min_{u_k}[G(x_k, u_k, k) + f_{k+1}(x_{k+1})]}$$

投资分配问题(纯离散问题)

- 某公司计划用40万元投资项目A, B, C. 下表给出了不同投资规模下的预期利润. 试制定最优投资计划

A			B				C			
1	2	3	1	2	3	4	1	2	3	4
20	30	40	10	20	30	40	10	20	30	40
1.8	2.8	3.2	1.2	1.9	2.5	3	0.8	1.6	2.4	3.1

建模

阶段? → 投资顺序

状态? → 剩余金额

决策? → 投资额

转移方程? → $x_{k+1}=x_k-u_k$

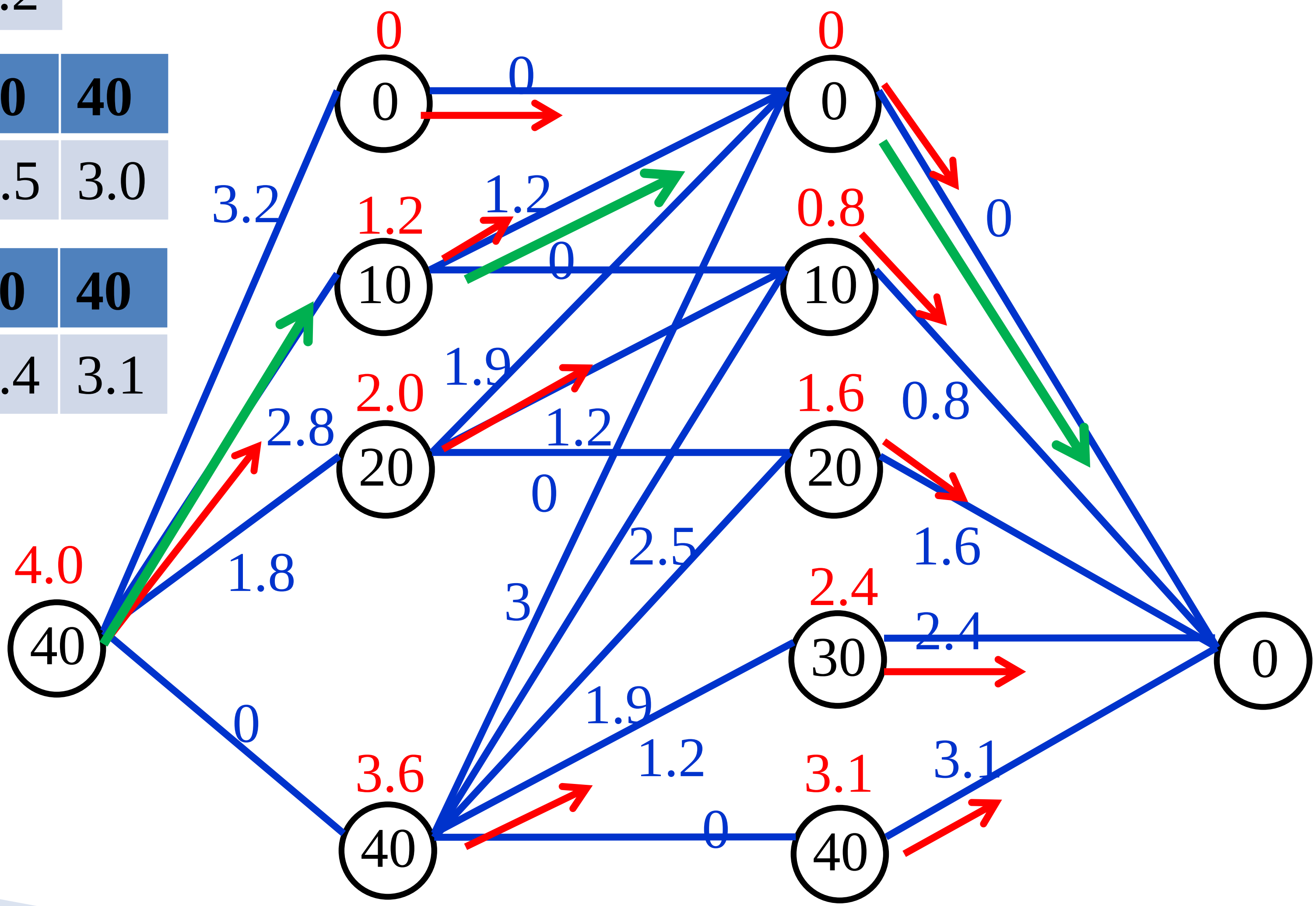
20	30	40
1.8	2.8	3.2

10	20	30	40
1.2	1.9	2.5	3.0

10	20	30	40
0.8	1.6	2.4	3.1

最优投资方案：

A投30万，B投10万，C投0万元



确定性定期多阶段决策问题

例2：(旅行商问题，Traveling Salesman Problem，TSP)

有 $n+1$ 个城市，记为$v_0, v_1, \ldots, v_n$，一个推销员从 v_0 出发，遍访$v_1, \ldots, v_n$各恰好一次后再返回 v_0，已知从 v_i 到 v_j 的旅费(或路程长度、耗时等) 为 $d_{i,j}$，求最优路线安排。

解：怎样划分阶段？按自然时序，划分为$n+1$个阶段

怎样定义状态？ **状态：每个阶段/时刻系统所处的状况、态势**

状态(v_i, V)： **v_i为当前时刻所在城市，V为尚未经过的城市集合(V中不包含 v_0)** **思考：状态数目？**$O(2^n)$

无后效性？

决策$(v_i, V) \to \left(v_j, V \setminus \{v_j\}\right), v_j \in V$ **决策费用为 $d_{i,j}$**

思考：画状态转移图？ **应利用基本方程求解！**

确定性定期多阶段决策问题

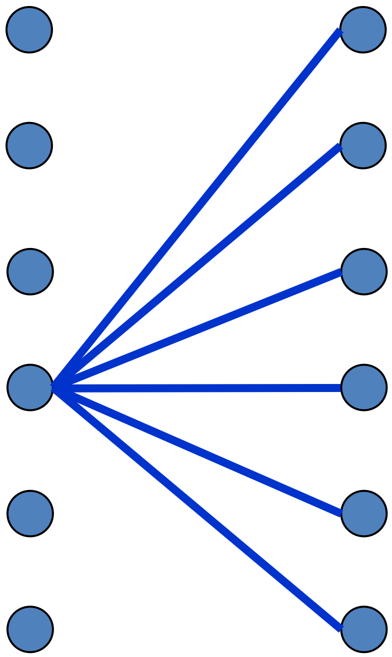
例2：(旅行商问题，Traveling Salesman Problem，TSP)

状态 (v_i, V)　　决策 $(v_i, V) \to \left(v_j, V \setminus \{v_j\}\right), v_j \in V$

怎样列基本方程？基本方程是关于cost-to-go的递推方程。

$f(v_i, V) = ?$ 从v_i出发，遍访V中所有城市各恰好一次，再回到 v_0的最短路程长度

状态转移图上求解过程的启示……



边界条件？

$$\begin{cases} f(v_i, \phi) = d_{i,0} \quad , \quad \forall v_i \neq v_0 \\ f(v_i, V) = \min\limits_{v_j \in V} \left\{ d_{i,j} + f\left(v_j, V \setminus \{v_j\}\right) \right\} \end{cases}$$

求 $f\left(v_0, \{v_1, v_2, \cdots, v_n\}\right) = ?$

确定性定期多阶段决策问题

例2：(旅行商问题，Traveling Salesman Problem，TSP)

状态 (v_i, V)　　决策 $(v_i, V) \to \left(v_j, V \setminus \{v_j\}\right), v_j \in V$

注意：非对称TSP

实例

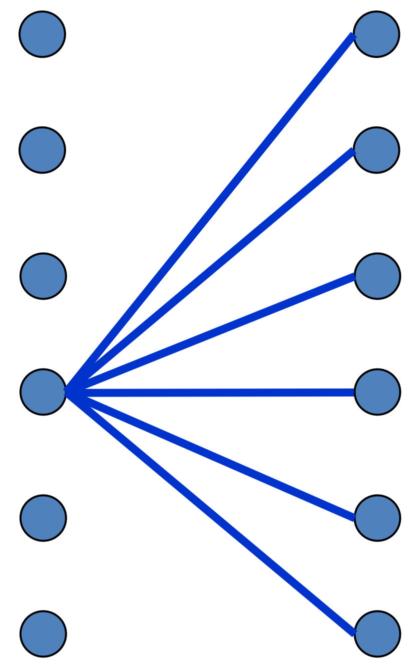
$$D = \begin{array}{c} \\ \\ \end{array}\begin{array}{cccc} v_0 & v_1 & v_2 & v_3 \end{array}$$

$$D = \left[\begin{array}{cccc} 0 & 8 & 5 & 6 \\ 6 & 0 & 8 & 5 \\ 7 & 9 & 0 & 5 \\ 9 & 7 & 8 & 0 \end{array}\right]\begin{array}{c} v_0 \\ v_1 \\ v_2 \\ v_3 \end{array}$$

最优解：$v_0 \to v_2 \to v_3 \to v_1 \to v_0$

P170~171

计算复杂性分析



$$\begin{cases} f(v_i, \phi) = d_{i,0} \quad , \quad \forall v_i \neq v_0 \\ f(v_i, V) = \min\limits_{v_j \in V} \left\{ d_{i,j} + f\left(v_j, V \setminus \{v_j\}\right) \right\} \end{cases}$$

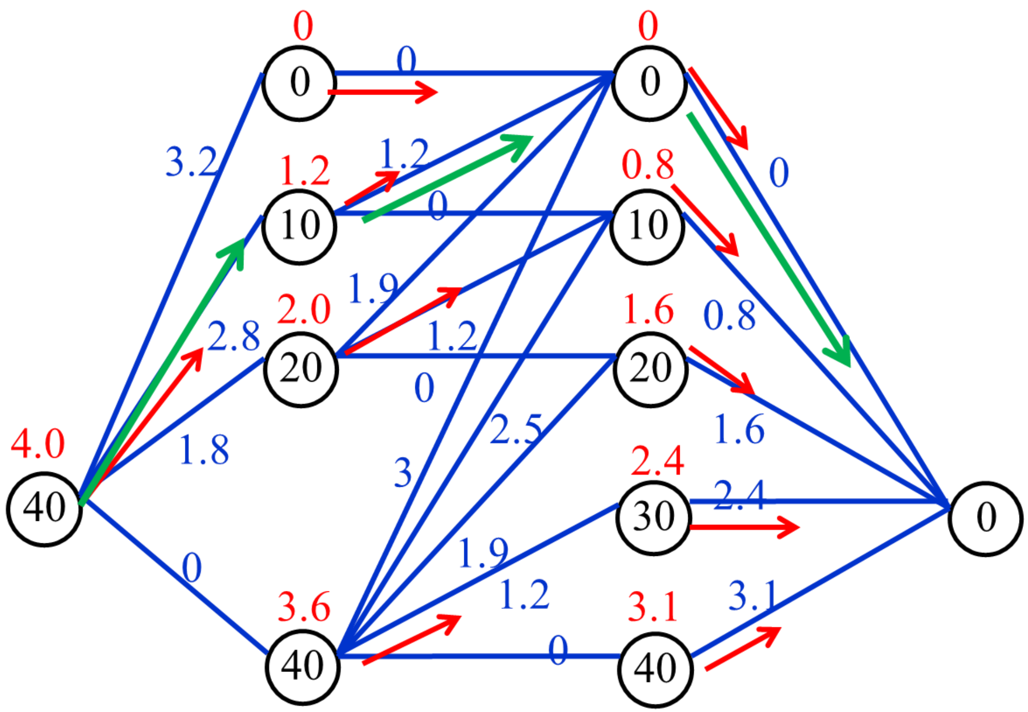
求 $f\left(v_0, \{v_1, v_2, \cdots, v_n\}\right) = ?$

动态规划 v.s. 非线性(混合整数)规划

- 确定性定期多阶段决策问题基本上都可以转化为非线性(混合整数)规划问题.
- 非线性(混合整数)规划问题转化为DP:
  - 最优化原理
  - 无后效性
  - 子问题的重叠性
- DP求解的原因
  - 全局解v.s.局部解
  - 中间信息
  - 求解效率

基本递推方程

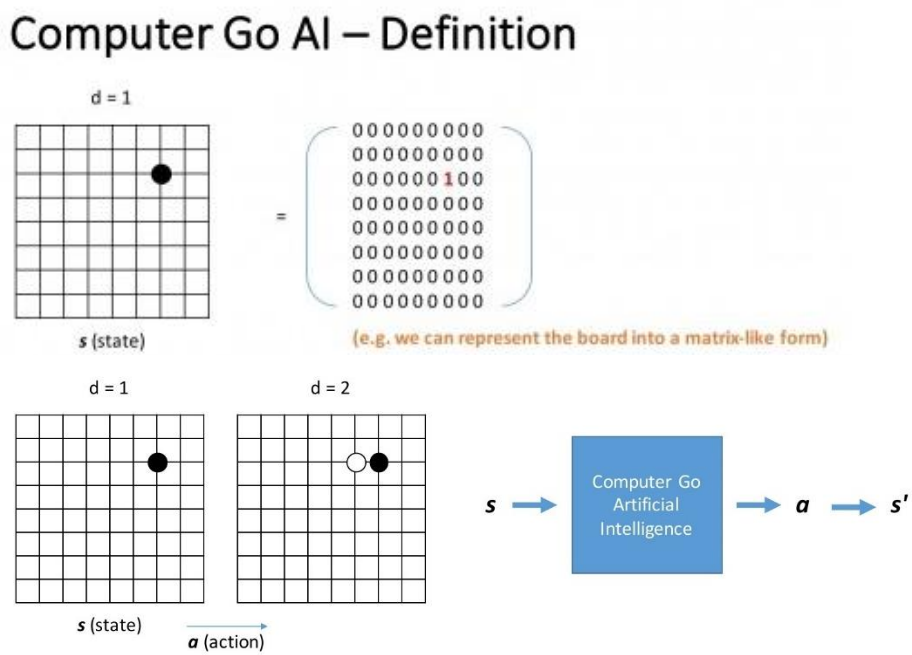
$$f_k(x_k) = \min_{u_k}[G(x_k, u_k, k) + f_{k+1}(x_{k+1})]$$



$$\pi^*(s) = \arg_a \min[r(s,a) + V^*(\delta(s,a))]$$

Mastering the Game of Go with Deep Neural Networks and Tree Search

(Nature 529, 484–489, 28 January 2016)



$$\boxed{\pi^*(s) = \arg_a \max[r(s, a) + V^*(\delta(s, a))]}$$

$|S|=3^{361}$ $|A_k|=361-2(k-1)$ $|\Omega|=361*359*357*\ldots..$

Reducing "act candidates"

Current Board

00 000 0000
00 000 1000
0 -1 0 0 1 -1 1 0 0
0 1 0 0 1 -1 0 0 0
00 00 -1 0000
00 000 0000
0 -1 000 0000
00 000 0000

Prediction Model

Next Action

000000000
000000000
000000000
000000000
000001000
000000000
000000000
000000000

s *f*: *s* → *a* *a*

Expert Moves Imitator Model (w/ CNN)

30,000,000 < s , a >

Updated Model ver 1.3 vs Updated Model ver 1.7

30,000,000 < s , a >

Current Board

00 000 0000
00 000 1000
0 -1 0 0 1 -1 1 0 0
0 1 0 0 1 -1 0 0 0
00 00 -1 0000
00 000 0000
0 -1 000 0000
00 000 0000

Deep Learning (13 Layer CNN)

000000 000
000000 000
000000 000
00000 0.2 0.1 00
00000 0.4 0.2 00
00000 0.1 000
000000 000
000000 000

Next Action

000000000
000000000
000000000
000000000
000001000
000000000
000000000
000000000

s *g*: *s* → p(*a*|*s*) p(*a*|*s*) argmax *a*

Reducing "act candidates"

ICLR 2015

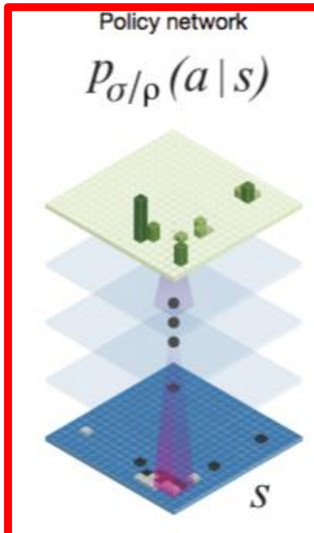
Feature	Planes	Description
Black / white / empty	3	Stone colour
Liberties	4	Number of liberties (empty adjacent points)
Liberties after move	6	Number of liberties after this move is played
Legality	1	Whether point is legal for cu
Turns since	5	How many turns since a mov
Capture size	7	How many opponent stones
Ladder move	1	Whether a move at this point
KGS rank	9	Rank of current player

Nature 2016

Extended Data Table 2 | Input features for neural networks

Feature	# of planes	Description
Stone colour	3	Player stone / opponent stone / empty
Ones	1	A constant plane filled with 1
Turns since	8	How many turns since a move was played
Liberties	8	Number of liberties (empty adjacent points)
Capture size	8	How many opponent stones would be captured
Self-atari size	8	How many of own stones would be captured
Liberties after move	8	Number of liberties after this move is played
Ladder capture	1	Whether a move at this point is a successful ladder capture
Ladder escape	1	Whether a move at this point is a successful ladder escape
Sensibleness	1	Whether a move is legal and does not fill its own eyes
Zeros	1	A constant plane filled with 0
Player color	1	Whether current player is black

Feature planes used by the policy network (all but last feature) and value network (all features).



Current Board

00 000 0000
00 000 1000
0 -1 0 0 1 -1 1 0 0
0 1 0 0 1 -1 0 0 0
00 00 -1 0000
00 000 0000
0 -1 000 0000
00 000 0000

Deep Learning (13 Layer CNN)

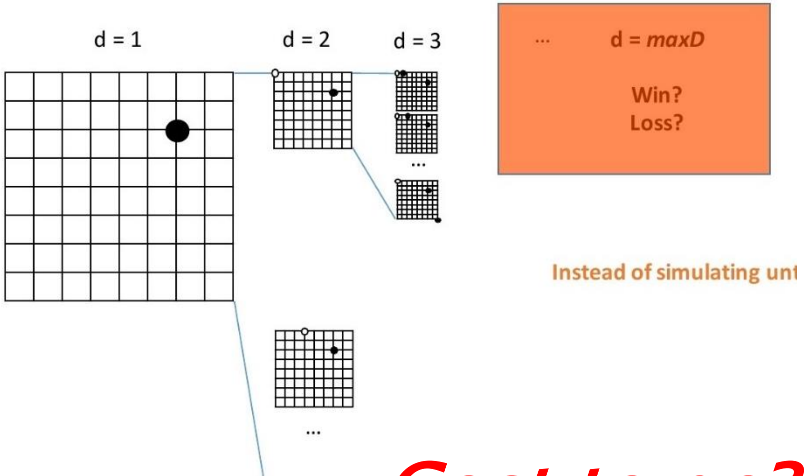
000000 000
000000 000
000000 000
000000.2 0.100
000000.4 0.200
000000.1 000
000000 000
000000 000

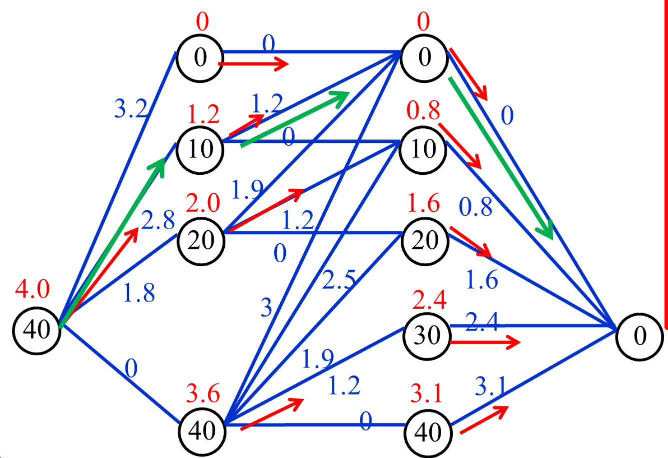
Next Action

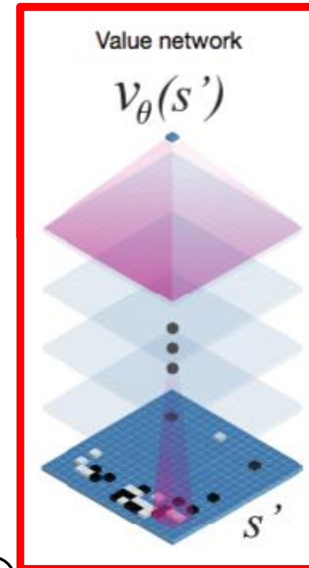
000000000
000000000
000000000
000000000
000001000
000000000
000000000
000000000

s $g: s \rightarrow p(a|s)$ $p(a|s)$ argmax a

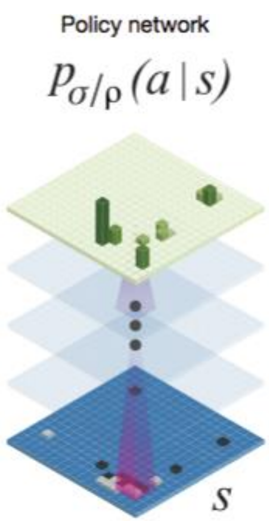
Board Evaluation



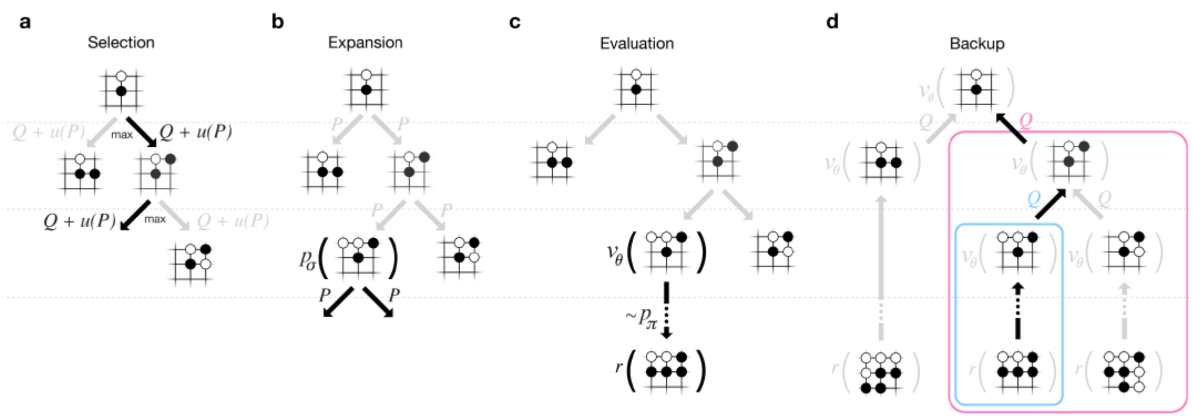




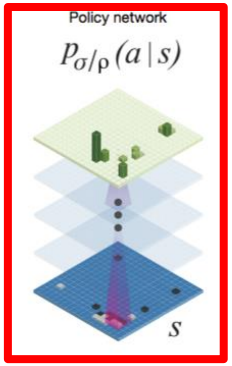
Cost to go?



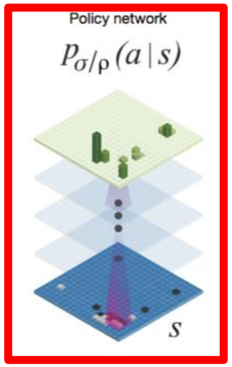
Monte-Carlo tree search



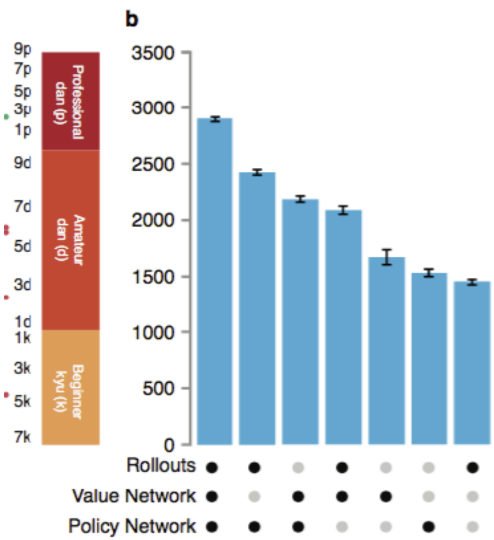
How AlphaGo selected its move

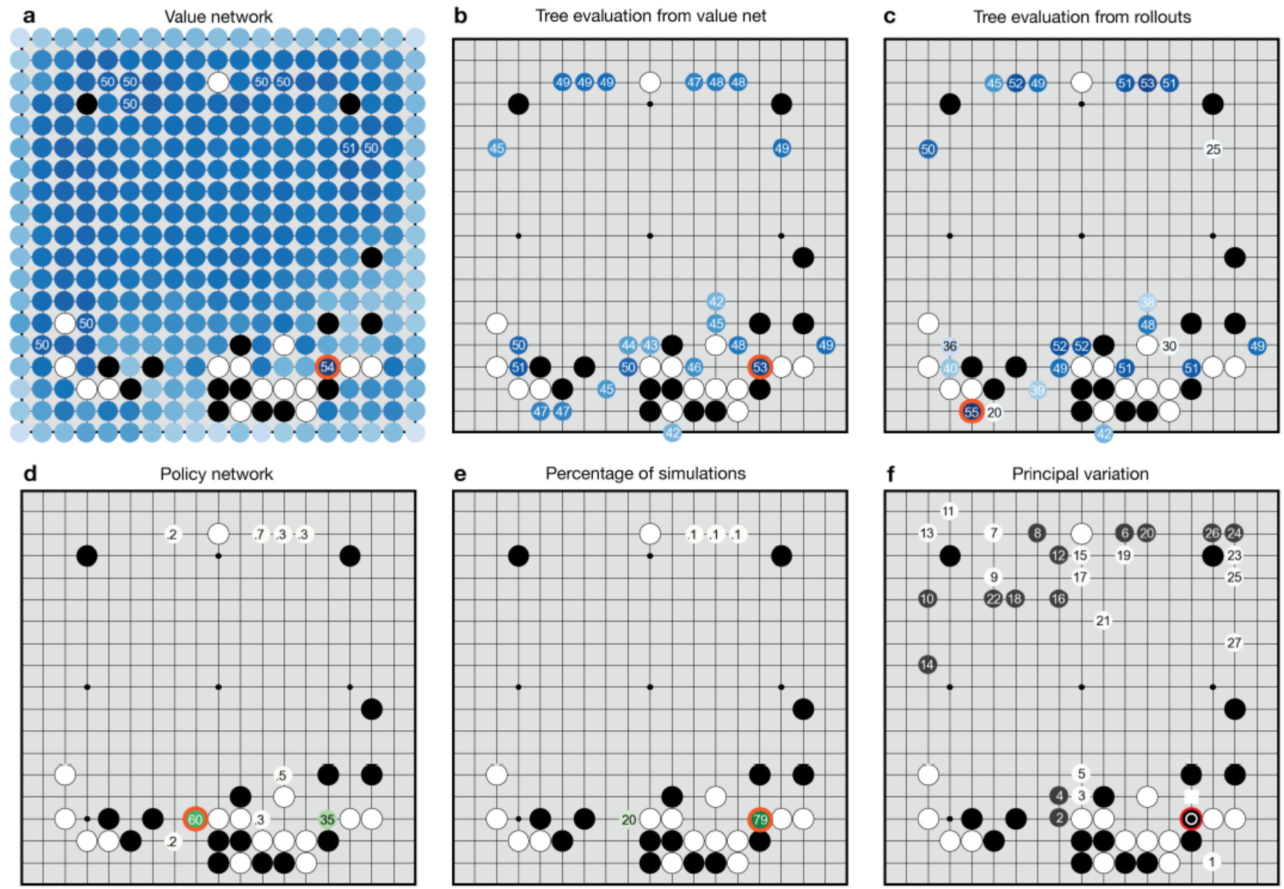


Bread reduction



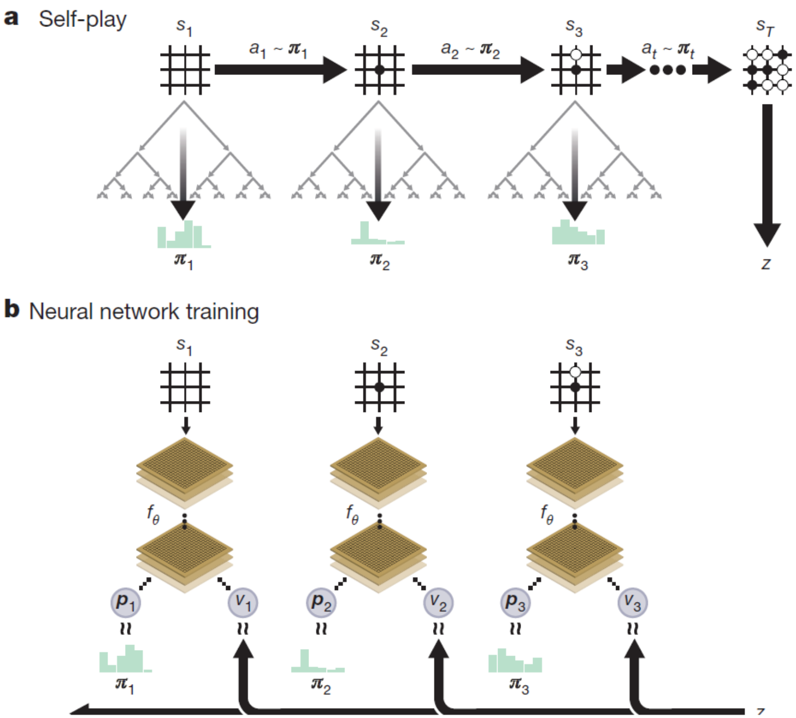
Depth reduction

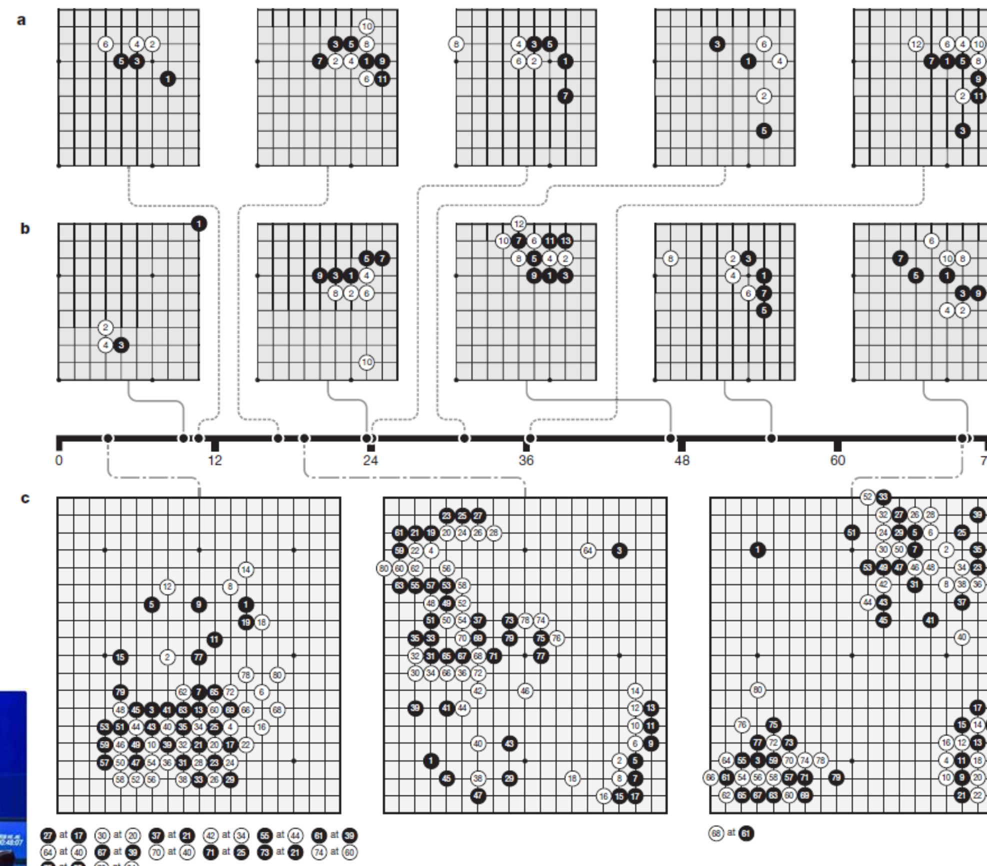




Nature 2017: Mastering the game of Go without human knowledge

1. without any human data
2. only stones as input features
3. single neural network
4. without any Monte Carlo rollouts





Challenges of Real-World Reinforcement Learning ---- ICML 19'

1	Training off-line from the fixed logs of an external behavior policy.
2	Learning on the real system from limited samples.
3	High-dimensional continuous state and action spaces.
4	Safety constraints that should never or at least rarely be violated.
5	Tasks that may be partially observable, alternatively viewed as non-stationary or stochastic.
6	Reward functions that are unspecified, multi-objective, or risk-sensitive.
7	System operators who desire explainable policies and actions.
8	Inference that must happen in real-time at the control frequency of the system.
9	Large and/or unknown delays in the system actuators, sensors, or rewards.