Изкуствен интелект—Зимен семестър 2001/2002. Преподавател: Иван А. Держански.

Лекция 4: Игри и тяхното моделиране

1 Връзка с ИИ

- Добре дефинирани задачи, изискващи интелигентност.
- Трудни задачи:
  - решават се в реално време (особено в турнирни условия);
  - предполагат неопределеност (не може да се предскаже ходът на противника);
  - пространството на състоянията в повечето случаи е огромно (напр. шах: разклоненост $= 35$, дълбочина $= 50$ двойни хода, 10^{40} различни възможни позиции, 35^{100} възела в дървото).
- Въпреки голямото пространство на търсене хората могат да се научат да играят добре.
- Подходяща област за изследване на методите за търсене.

Класически тип игри, разглеждани в ИИ: абстрактни игри за двама играчи с пълна информация, със или без случаен елемент. Най-разработваните игри са шахът и шашките.

2 Формализация на задачата

- Частен случай на задачата за търсене в пространството на състоянията.
- Състоянията, в които играта свършва, се наричат *терминални*. Краят на играта може да е равен (реми) или да означава победа за единия играч (и съответно загуба за другия).
- Всяко терминално състояние има определена стойност (например победа на първия играч $= 1$, загуба $= -1$, реми $= 0$).
- Дървото на търсене е разделено на нива, на всяко от които инициативата принадлежи на единия играч: Макс, стремящ се да максимизира стойността на състоянието, и Мини, целяща да го минимизира.

3 Минимаксен алгоритъм

- Предполагаме, че имаме пълното дърво на играта до терминалните възли. (Това е практически възможно само за най-елементарните игри като кръстчета и нулички или хексапон.)
- Стойността на възлите се изчислява отдолу нагоре, започвайки от терминалите.
- За всеки възел на Макс се избира максималната стойност на неговите наследници (т. е. ходът, който максимизира стойността на състоянието).
- За всеки възел на Мини се избира минималната стойност на неговите наследници (т. е. ходът, който минимизира стойността на състоянието).
- По този начин се определя оптималният ход в корена на дървото.

4 Евристично оценяване на състоянието

- Отсичане на дървото от определен възел надолу и намиране на евристична оценка на състоянието.

- Идеалната евристика би трябвало да отразява вероятността, че Макс ще победи (респ. Мини ще загуби) при дадената позиция. При добре познатите игри се използува опитът и интуицията на добрите играчи. Евристиките могат да бъдат усъвършенствувани и експериментално.

- Пример за елементарна евристика при шаха: разлика от тегловните суми от броя на белите и черните фигури (при което напр. 1 пешка е равна на $\frac{1}{3}$ кон или офицер, $\frac{1}{5}$ топ или $\frac{1}{9}$ дама). Разположението на фигурите може да бъде отчетено чрез допълнителни коефициенти.

- Определяне на нивото на отсичане (предполага се, че евристиките за по-долните нива на дървото се намират по лесно и са по-точни):

  - Предварително задаване на граница на дълбочината.
  - Използуване на итеративно търсене по нива при предварително зададено време за изчисление.
  - Търсене на затихване (quiescence search): развиване на дървото до достигане на "спокойна" позиция (позиция, която не предполага рязка промяна на оценката в непосредственото бъдеще).
  - Проблем на хоризонта (horizon problem): събития, съществено променящи стойността на състоянието, които не могат да се предотвратят, но могат да се отлагат неопределено дълго.

5 Подобряване на ефективността

(В шахматните турнири играчът има 150 s за обмисляне на всеки ход. Ако се разглеждат по 1000 позиции на секунда, това значи 150 000 позиции, или 3–4 хода—ниво на начинаещ играч.)

Съкращаване на броя на перспективните възли: кастрене (отсичане) алфа–бета (α–β pruning или α–β cutoff).

- Два текущи параметъра по време на търсене в дървото на играта:

  - α: най-голямата получена досега стойност на възлите на Макс;
  - β: най-малката получена досега стойност на възлите на Мини.

- Ако възел на Мини има стойност, по-голяма от (или равна на) β, неговите съседи на същото ниво могат да не се оценяват.

- Ако възел на Макс има стойност, по-малка от (или равна на) α, неговите съседи на същото ниво могат да се пренебрегнат.

- Ефективността на метода зависи от реда на обхождане на наследниците на позициите в дървото. Ако най-перспективните ходове се разглеждат първи, се получава намаляване на сложността по време от $O(b^d)$ до $O(b^{\frac{d}{2}})$, т. е. за същото време може да се разгледат два пъти повече ходове напред.

  Ако се прилага итеративно задълбочаване, стойностите от предишната итерация могат да се запомнят и да определят реда, в който се развиват при следващата.

6 Игри със случаен елемент

- Класически пример: табла.
- Освен нивата на Макс и Мини в дървото присъстват и нива на случая (на заровете), броят на възлите в които зависи от възможните изходи на случайното събитие (21 при хвърляне на два зара).
- Един вариант на минимаксния алгоритъм, включващ случайност ($f(N)$ е стойността на ситуацията във възел N):
  - d_i $(i = 1, \ldots, n)$: различните изходи от случайното ниво.
  - $P(d_i)$: вероятност за случаен изход d_i.
  - $S(N, d_i)$: множества ходове от позиция N за всеки случаен изход d_i.
  - Ако в N е на ход Макс, $f(N) = \sum_{i=1}^{n} p(d_i) \max_{s \in S(N,d_i)} f(s)$.
  - Ако в N е на ход Мини, $f(N) = \sum_{i=1}^{n} p(d_i) \min_{s \in S(N,d_i)} f(s)$.
- Възможните стойности на състоянията не са само измежду стойностите на терминалите. Следователно важна е не само наредбата между терминалите (печалба, загуба, равен), а и абсолютната им стойност.
- Нарастване на сложността по време: $O(b^d n^d)$ (при n случайни изхода). На практика рядко се използва голяма дълбочина—пределът при таблата е 2 двойни хода.
- Поради наличието на случаен елемент кастренето алфа–бета не е непосредствено приложимо. То може да бъде адаптирано, но ефективността му е по-ниска.

Drew McDermott: *How Intelligent is Deep Blue?*
`http://www.cs.yale.edu/HTML/YALE/CS/homepage/op-ed/deep_blue.html`