

Лекция 3:

Търсене в пространството на състоянията (продължение)

АНАЛИЗ НА ЦЕЛИТЕ И СРЕДСТВАТА

General Problem Solver (GPS)

GPS (General Problem Solver, Обща програма за решаване на задачи) е ранна програма за търсене, която има изключително влияние върху изследванията по ИИ. Тя реализира алгоритъм за търсене, който използва функция за оценка на състоянията и функция за нареждане на операторите, за да направлява търсенето. Следователно, тя не прилага всички възможни оператори към дадено състояние, а само тези, които изглеждат перспективни.

GPS е била предназначена да моделира човешкото поведение в задачите за търсене като игрови задачи, символно интегриране и др. Целта ѝ е била прекалено амбициозна (не всички проблеми могат да се разглеждат като задачи за търсене), но при някои предположения GPS може да се използва за генериране на планове на действия, за решаване на задачи и самообучение по аналогия и др.

GPS прилага една техника, наречена **анализ на целите и средствата** (Means-Ends Analysis, MEA). Най-общо MEA е методология за търсене в пространството на състоянията, при която на всяка стъпка се прави опит за редуциране на различията между текущото състояние и целта.

Means-Ends Analysis (MEA)

Идеята на MEA се състои в следното. При условие, че е дадено състоянието, което трябва да се преобразува, то се **съпоставя** с описанието на целевото състояние. В резултат на съпоставянето или се установява, че няма разлика между текущото и целевото състояние (и следователно текущото състояние съвпада с целевото състояние), или се намират съществуващите разлики между текущото и целевото състояние. Във втория случай GPS се опитва да намали разликите между текущото и целевото състояние.

GPS използва разликите между състоянията по два начина. Първо, броят на разликите служи като груба мярка за общия напредък към целта. Второ, разликите могат да предложат какви действия да се предприемат по-нататък.

Всъщност преброяването на разликите е само груба мярка за разстоянието до целта, защото то пренебрегва факта, че някои разлики са по-важни от други. Затова обикновено се въвеждат тегла (приоритети) на разликите, които намират отражение във формулировката на оценяващата функция.

Втората употреба на разликите в MEA е за направляване на избора на операторите. Досега операторите се третираха като “черни кутии”, към които се обръщаме, за да генерираме 0, 1 или повече наследници на дадено състояние. В GPS операторите имат по 3 компонента:

- предусловия;
- трансформираща функция;
- списък на намаляваните разлики.

Първият компонент представлява описание (в същите термини, както и описанието на целевото състояние), което включва необходимите (а често и достатъчни) условия за приложимост на оператора.

Вторият компонент представлява “черната кутия”, която генерира наследниците на разглежданото състояние.

Третият компонент свързва оператора с разликите, които той редуцира (за чието намаляване той е подходящ). Обикновено този компонент се съхранява в специална структура, наречена *таблица на връзките* или *таблица оператор–разлика*. Тази таблица се предоставя от потребителя (от поставящия задачата) и позволява на GPS при дадена разлика да намери оператор за нейното намаляване.

Алгоритъм MEA(Current,Goal)

1. Сравняват се (съпоставят се) Current и Goal. Ако между тях няма разлики, тогава алгоритъмът прекратява работата си.

2. В противен случай се избира най-важната разлика между Current и Goal. Избраната разлика се редуцира чрез изпълнение на следните действия (докато се стигне до успех или неуспех):

(а) Избира се неизползван (неизпробван) до момента оператор O, който е приложим към избраната разлика. Ако не съществуват такива оператори, алгоритъмът завършва с неуспех.

(б) Прави се опит да се приложи O към Current. Генерират се описанията на две състояния:

O-Start (състояние, в което са изпълнени предусловията на O) и

O-Result (състояние, което се получава след прилагането на O върху O-Start).

(в) Ако

First-Part := MEA(Current,O-Start) и

Last-Part := MEA(O-Result,Goal)

завършат с успех, то алгоритъмът завършва с успех и връща като резултат конкатенацията на First-Part, O и Last-Part.

ТЪРСЕНЕ НА ЦЕЛ ПРИ СПАЗВАНЕ НА ОГРАНИЧИТЕЛНИ УСЛОВИЯ (Constraint Satisfaction Problems, CSP)

Формулировка

Дадени са:

- множество от *променливи* v_1, v_2, \dots, v_n (с области на допустимите стойности)
- множество от *ограничения* (допустими/недопустими комбинации от стойности на променливите)

Целево състояние (състояния): множество от стойности на променливите $\{v_1=c_1, v_2=c_2, \dots, v_n=c_n\}$, които не нарушават ограниченията.

Примери

Задача за осемте царици, решаване на криптограми, оцветяване на географска карта, съставяне на разписания и др.

Алгоритми

- генериране и тестване (generate and test)
- търсене с възврат (backtracking)
- разпространяване на ограниченията (forward checking)
- предварително проучване (look ahead)

МОДЕЛИРАНЕ НА ИГРИ

Разглеждат се т. нар. интелектуални *игри с пълна информация*, които се играят от двама играчи и върху хода на които не оказват влияние случайни фактори. Двамата играчи играят последователно и всеки от тях има пълна информация за хода на играта.

Най-често се решава задачата за намиране на най-добър първи ход на играча, който трябва да направи текущия ход.

ДС при тези задачи е дърво на възможните позиции в резултат на възможните ходове на двамата играчи. Общият брой възли в ДС е от порядъка на b^d (b - коефициент на разклонение на игровото дърво, d – височина (дълбочина) на игровото дърво).

Пример. В шахмата е установено, че b има средна стойност около 35, а всеки играч играе около 50 хода, т.е. d има средна стойност около 100.

Минимаксна процедура

Обща характеристика

Метод за намиране на най-добрия ход на първия играч при предположение, че другият играч също играе оптимално. Изисква построяване на цялото ДС и намиране на оценките на листата (наричат се *статични оценки*).

Предполага се, че двамата играчи имат противоположни интереси, изразени в това, че единият търси стратегия, която води до получаване на позиция с максимална оценка (максимизиращ играч), а другият търси стратегия, която води до получаване на позиция с минимална оценка (минимизиращ играч).

Минимаксната процедура е метод за получаване на оценките (наричат се *придобити* или *породени оценки*) на възлите от по-горните нива на ДС, които позволяват на първия играч да избере най-добрия си ход.

Получаване на оценките на възлите от ДС

Оценките се разпространяват отдолу нагоре, като всеки от възлите, съответни на ход на максимизиращия играч, получава оценка, равна на максималната от оценките на преките му наследници, а всеки от възлите, съответни на ход на минимизиращия играч, получава оценка, равна на минималната от оценките на преките му наследници.

Оценка на минимаксната процедура

Изисква построяване на цялото ДС, което може да се окаже твърде голямо в общия случай на реална игра. Точен, но крайно неефективен и често нереализуем на практика метод.

Алфа-бета процедура

Идея на метода

Преодоляват се проблемите при минимаксната процедура, породени от обстоятелството, че там процесът на генерирането на ДС е напълно отделен от процеса на оценяване на позициите, което води до силна неефективност. Тук листата се оценяват веднага след генерирането им и при първа възможност се пресмятат и съответните придобити (породени) оценки на възлите от по-горните нива или поне се намират подходящи горни граници на оценките на възлите от минимизиращите нива (съответни на ходове на минимизиращия играч) и/или долни граници на оценките на възлите от максимизиращите нива (съответни на ходове на максимизиращия играч). Така броят на операциите за намиране на същия резултат, както при минимаксната процедура, може да се намали значително.

Алфа-бета процедурата изисква генериране на ДС в дълбочина, при което се преценява безполезността от генерирането и оценяването на някои клонове, не оказващи влияние върху резултата.

Дефиниции

Алфа-стойност на даден максимизиращ възел се нарича текущо установената долна граница на породената му оценка. *Бета-стойност* на даден минимизиращ възел се нарича текущо установената горна граница на породената му оценка.

Първоначално определените алфа- и бета- стойности на възлите могат да се уточняват в процеса на работа, като алфа-стойностите могат само да растат, а бета-стойностите – само да намаляват.

Описание на метода

Правила за прекратяване на генерирането и търсенето:

- *Алфа-отсичане*. Не е необходимо да се извършва генериране и търсене върху всяко поддърво, което произлиза от минимизиращ възел, бета-стойността на който е по-малка или равна на алфа-стойността на съответния максимизиращ родител.
- *Бета-отсичане*. Не е необходимо да се извършва генериране и търсене върху всяко поддърво, което произлиза от максимизиращ възел, алфа-стойността на който е по-голяма или равна на бета-стойността на съответния минимизиращ родител.

Оценка на алфа-бета процедурата

При използване на алфа-бета процедурата се получава напълно точен резултат.

Броят на генерираните и оценени възли при алфа-бета процедурата е най-малко от порядъка на $b^{d/2}$. Тази стойност се получава в идеалния случай, когато оценките на листата са наредени максимално добре. Следователно алфа-бета процедурата намалява скоростта на развитие на комбинаторния взрив, но не го предотвратява.

Практически реализации на минимаксната процедура

- Отсичане на игровото дърво от дадено ниво надолу и намиране на евристични оценки на получените терминални възли (които в общия случай не са листа в ДС).
- Идеалната евристика би трябвало да отразява вероятността, че съответният играч ще победи в дадената (оценяваната) позиция. При добре познатите игри се използват опитът и интуицията на силните играчи. Евристиките могат да бъдат усъвършенствани експериментално.
- *Пример* за елементарна евристика при шахмата: разлика на тегловните суми от броя на белите и черните фигури (при което например 1 пешка е равна на 1/3 кон или офицер, 1/5 топ или 1/9 дама). Разположението на фигурите може да бъде отчетено чрез специално подбрани коефициенти.
- Определяне на нивото на отсичане (предполага се, че евристичните оценки на възлите от по-долните нива се намират по-лесно и са по-точни):
 - Предварително задаване на горна граница на дълбочината на генерираната част от ДС;
 - Използване на итеративно търсене по нива при предварително зададено максимално време за работа;
 - Търсене на затихване: генериране на дървото до достигане на “спокойна” позиция (позиция, която не предполага рязка промяна на оценката в непосредствено бъдеще).