DETERMINISTIC GAMES PLAYING WITH LEARNING

Jakub Knoflíček

Bachelor Degree Programme (3), FIT BUT

E-mail: xknofl00@stud.fit.vutbr.cz

Supervised by: František Vítězslav Zbořil

E-mail: zboril@fit.vutbr.cz

Abstract: This paper deals with artificial intelligence for computer player for deterministic games such as checkers. It summarizes ideas and methods for best move searching for a current game state in very large state space. Then temporal difference learning method is presented which helps us to evaluate game states. These mechanisms are used in chess playing program.

Keywords: Reinforcement learnining, temporal difference (TD) learning, alfa-beta pruning, checkers

1. ÚVOD

Cílem mé práce bylo zaměřit se na uplatnění posilovaného učení v prostředí deterministických her. Za takové hry můžeme považovat například šachy, vrchcáby či dámu, která byla vybrána k ověření použitelnosti a úspěšnosti použitých metod. Technika posilovaného učení stejně jako algoritmus minimax rozšířený o alfa-beta řezy spadají do oboru umělé inteligence a slouží nám k volbě, pokud možno, nejlepšího tahu v daném stavu hry. Počet možných stavů dané hry, které následují v dalších tazích, jež je program schopen projít a vhodně ohodnotit, je klíčem k úspěšné simulaci lidského protihráče počítačovým programem. Například projekt CHINOOK v roce 2007 byl schopen pracovat s databází s 39 biliony možných stavů hry dáma [1].

2. ROZBOR

2.1. METODA MINI-MAX

Tato metoda je klasickým algoritmem pro hraní tzv. složitých deterministických her (tj. her, u kterých nelze v reálném čase projít celý stavový prostor). Základem je jednoduchá logická myšlenka, že hráč, který je aktuálně na tahu, vybírá takový tah, který vede k pro něj nejlepšímu (nejlépe hodnocenému) stavu. Naopak protihráč aktuálního hráče se bude snažit v dalším kroku vybrat tah, který vede ke stavu nejméně výhodnému pro aktuálního hráče. Tyto dvě varianty se dále opakují. Proto bývá metoda nazývána Mini-Max. Prohledávaný prostor má obecně podobu stromové struktury. Jak jsem se zmínil v úvodu, důležitým faktorem je rozsah prohledávaného stavového prostoru, který je zadán maximální hloubkou prohledávání v tomto stromu. Obecně se algoritmus rekurzivně zanořuje, dokud nenarazí na koncový stav hry, či nedosáhne zadané hloubky zanoření. Tento algoritmus ovšem nezohledňuje, že v určitých situacích nemá význam pokračovat v prozkoumávání dané části stavového prostoru, jelikož se již ví, že nebude nikdy vybrána. Proto bývá algoritmus často doplněn Alfa-Beta řezy, které tuto skutečnost zohledňují.

Někdy je tato metoda mylně chápána ve smyslu, že určuje průběh další části hry. Zdůrazněme, že slouží pouze k výběru jednoho tahu v daném stavu hry.

2.2. ALFA-BETA ŘEZY

Alfa-Beta řezy jsou rozšířením metody Mini-Max a někdy bývá toto spojení za samostatnou metodu Alfa-Beta. Při průchodu stromovou strukturou reprezentující danou část stavového prostoru jsou předávány také dvě proměnné s určitými hodnotami (Alfa a Beta). Při hledání maximálního ohod-

nocení je hodnota uložena do proměnné Alfa, jinak do proměnné Beta. V okamžiku, kdy není Alfa menší než Beta, nemá význam pokračovat v prohledávání dané části podstromu. Tímto opatřením lze často dosáhnout až o cca třetinu menšího počtu procházených stavů, což je hlavním přínosem Alfa-beta řezů. Podrobnější princip lze též nalézt v [3]. Ušetřený čas lze věnovat například pro zvětšení prohledávané hloubky, čímž bude moci program nalézt tah, který nakonec vede k lépe ohodnocenému stavu hry, popřípadě k vítězství.

2.3. POSILOVANÉ UČENÍ METODOU TD-LEARNING

Největší přínos do rozhodujícícího procesu přináší posilované učení, konkrétně mnou použitá metoda TD-learning. Zkratkou TD je míněno temporal differnece, což je chápáno jako rozdíl po sobě jdoucích předpovědí [2].

Spojení Alfa-Beta řezů s metodou Mini-Max nám umožní výběr vhodného tahu, ovšem vlastní ohodnocení aktuálního stavu hry neprovádí. O to se stará tzv. hodnotící funkce. Pokud bychom měli funkci, která stavu okamžitě přiřadí nejvhodnější ohodnocení, bylo by to nejlepším řešením. Ale nalezení takové funkce je v praxi velmi složité, v mnohých případech nemožné. Užijeme-li principů posilovaného učení, například metody TD-learning, budeme mít na počátku program, který považuje všechny tahy za stejně dobré(vybírá náhodný tah), ale s počtem odehraných her se jeho schopnosti zlepší a bude schopen lépe ohodnotit jednotlivé stavy hry. Proto je vhodné program nechat nejprve udělat sadu tzv. náhodných procházek, kdy se bude učit, a teprve poté proti němu nechat hrát lidského hráče. Na rovnici **1** vidíme princip ohodnocení metodou TD-learning. Nejpve jsou známy hodnoty pouze konečných stavů hry. Od nich se postupně hodnotí stavy směrem k počátečnímu stavu hry. Vždy je přepočítána hodnota aktuálního stavu na základě hodnoty stavu následujícího.

$$U^{\pi}(s) = U^{\pi}(s) + \alpha\left(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s)\right) \qquad (1)$$

Vztah si poněkud přibližme. $U^{\pi}(s)$ nám vyjadřuje ohodnocení aktuálního stavu s. Obdobně $U^{\pi}(s')$ vyjadřuje ohodnocení následujícího stavu s'. Faktor γ udává, jak mnoho ovlivňuje hodnota dalšího stavu ohodnocení stavu aktuálního. $R(s)$ definuje odměnu za dosažení statvu s. V mém programu je odměna udělena pouze za dosažení jednoho z koncových stavů hry a podle toho, zda se jedná o výhru či prohru uděluji maximální, či naopak minimální ohodnocení. Nakonec je velmi důležitý koeficient učení α. Nastavil jsem jeho hodnotu jako dynamickou, která se s počtem návštěv daného stavu snižuje. Toto nám zajistí, že hodnota určitého stavu postupně konverguje k poměrně stálé hodnotě. Celý postup přehodnocování stavů je vidět na **Algoritmus 1**. Charakteristickou vlastností metody TD-learning je to, že umožňuje přepočítávat ohodnocení stavu po odehrání jednoho herního tahu (ne až po celé jedné hře).

```
funkce ohodnoceni(aktualní stav, nový stav){
    konstanta faktor γ, udávající vliv ohodnocení nového
    stavu na aktuální stav
    zjisti aktualní ohodnocení aktualního stavu
    zjisti aktualní ohodnocení nového stavu
    zjisti počet návštěv nového stavu
    zjisti hodnotu koeficientu α na základě počtu návštev
    nového stavu
    je-li stav jedním z koncových, zjisti hodnotu odměny
    vypočti hodnotu nového stavu dle metody TD-learning
    vrať nové ohodnocení aktuálního stavu
}
```

Algoritmus 1: Přehodnocení aktuálního stavu metodou TD-learning

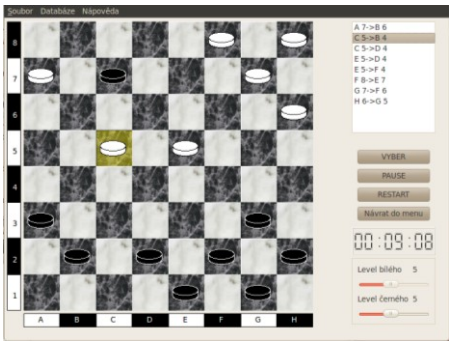
Poznamenejme, že počet všech možných stavů je výrazně velký (v řádu trilionů), proto není v schopnostech programu na běžném počítači ohodnotit všechny stavy, právě naopak. Je proto vhodné mít mechanismus pro určení náhradní hodnoty takového stavu.

3. IMPLEMENTACE

Implementace výpočetní části programu vychází z uvedených principů metody Alfa-Beta kombinovaného s učením TD-learning. Program bude postupně splňovat více možností správy možných herních stavů důležitých pro metodu posilovaného učení. Program bude moci načíst databázi ze souboru předem daného formátu a uložit informace o stavech do asociativního pole. Takto lze nejrychleji přistupovat k hledaným stavům a jejich hodnotám. Výpočet dalšího tahu je možné provádět ve zvláštním vláknu, tudíž není grafické vlákno pozastaveno.

Implementovány jsou dvě varianty hry dáma, česká a mezinárodní. To umožňuje sledovat náročnost procházení různě velkých stavových prostorů, rychlost generování odlišných pravidel a také délku hry.

Program je vyvíjen ve frameworku Qt verze 4.7, což přináší kromě podpory programování grafického prostředí programu, také mechanismus signálů a slotů. Tento je například využit při simulované hře počítače proti počítači, kdy je třeba po každém kroku chvíli počkat, aby bylo lidské oko schopno postřehnout, co se vlastně ve hře událo. Použití časovače spolu se signálem a přiděleným slotem nám umožňuje neblokující čekání. Také spojení jazyka C++ a tohoto frameworku umožňuje zachovat přenositelnost programu mezi běžnými operačními systémy a užít mechanismus vláken. Vzhled beta verze programu ukazuje **Obrázek 1**, který pochází z operačního systému Ubuntu.



Obrázek 1: Vzhled beta verze programu

4. ZÁVĚR

Po dokončení implementační části bude předmětem dalšího zkoumání hledání optimálního nastavení faktoru γ a koeficientu učení α ve funkci pro TD-learning v kombinaci s maximální hloubkou prohledávání stavového prostoru metodou Alfa-Beta, tak aby program byl schopen co nejlépe směřovat partii k výhře. Také bude hledán optimální mechanismus náhradního hodnocení stavů, které neohodnotil algoritmus TD-learning. Nakonec budou porovnány herní výsledky pro českou a mezinárodní variantu hry dáma.

REFERENCE

[1] Russell, S. a Norvig, P.: Artificial intelligence: A modern approach, New Jersey , Prentice Hall 2010, ISBN 978-0-13-207148-2.

[2] Tesauro, G.: Temporal Difference Learning and TD-Gammon, Communications of the ACM, č. 38.

[3] Edwards, D.J. a Hart, T.P.: The Alpha-Beta Heuristic, Technical report, Cambridge, MA, 1963