Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Václav Venta

MCTS pro hru Metro

Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: RNDr. Jan Hric

Studijní program: Informatika

Studijní obor: Obecná informatika

Praha 2013

Rád bych poděkoval všem, kteří mě podpořili při psaní této práce. Zvláštní poděkování patří RNDr. Janu Hricovi za přínosné konzultace a cenné rady.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 2.8.2013 Václav Venta

Název práce: MCTS pro hru Metro

Autor: Václav Venta

Katedra: Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: RNDr. Jan Hric

Abstrakt: Metoda MCTS byla poprvé představena v roce 2006 a okamžitě uspěla ve hře GO. Její největší výhodou je, že nepotřebuje ohodnocovací funkci, která se ve spoustě her často obtížně navrhuje. Místo toho je založena na náhodných simulacích. Tato práce popisuje principy metody MCTS a zabývá použitelností MCTS v deskové hře Metro. V textu jsou popsány vlastnosti hry Metro, díky kterým je pro metodu MCTS vhodná. Součástí práce je software Metro sloužící jako herní prostředí upravené Metro. V něm je naimplementována umělá inteligence založená na metodě MCTS.

Klíčová slova: MCTS, Metro, teorie her

Title: MCTS for the game Metro

Author: Václav Venta

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: RNDr. Jan Hric

Abstract: MCTS method was first introduced in 2006 and it immediately succeeded in the game of GO. Its biggest advantage is that it doesn't require any evaluation function, which is often hard to design in a lot of games. Instead it is based on randomized simulations. This thesis describes MCTS principles and deals with its suitability for the board game Metro. The text describes Metro characteristics which make the game suitable for the use of MCTS. A software serving as game environment is attached to this work on CD. It has artificial intelligence based on MCTS method implemented in it.

Keywords: MCTS, Metro, game theory

Obsah

Úvod

Tato bakalářská práce se zabývá použitelností metody Monte Carlo Tree Search (MCTS) pro hru Metro. Součástí práce bylo naprogramovat herní prostředí umožňující hru 2 i více hráčů, jejichž umělá inteligence je založená na metodě MCTS. Efektivitu naší implementace pak otestujeme pomocí experimentů.

Metoda MCTS se stává úspěšnou v logických hrách. Prakticky okamžitě po uvedení se prosadila ve hře Go. Výhodou této metody je, že nepotřebuje ohodnocovací funkci.

Metro je desková hra s úplnou informací, jejíž výhodou je její konečnost. To jsou vhodné předpoklady pro použití metody MCTS pro tuto hru.

První kapitola se zabývá hrou Metro. Jsou zde uvedeny základní informace o hře i její pravidla tak, aby i neznalý čtenář pochopil základní principy hry.

Druhá kapitola popisuje obecné principy metody MCTS a uvádí nejběžnější strategie používané v jednotlivých krocích algoritmu.

Ve třetí kapitole je rozebrána vhodnost metody MCTS pro použití ve hře Metro. Dále je zde popsána naše implementace MCTS.

Ve čtvrté kapitole jsou prezentovány výsledky našich experimentů. Je zde ukázána efektivita naší heuristiky a popsány vhodné hodnoty některých parametrů.

1 Hra Metro

V této kapitole si představíme hru Metro. Popíšeme si její principy a pravidla a nakonec rozebereme taktiku ve hře.

1.1 O hře

Deskovou hru Metro vytvořil v roce 1997 německý herní designer Dirk Henn. Inspiroval se událostmi z roku 1900, kdy se v Paříži konala jedna z největších světových výstav. Francouzská metropole se k této příležitosti rozhodla posílit svojí dopravní infrastrukturu a tak začala s výstavbou metra.

V roce 2000 získala hra nominaci na ocenění Spiel des Jahres za nejlepší hru roku, což je u moderních společenských her nejprestižnější ocenění. Sám Henn toto ocenění přirovnává k Oscaru mezi filmovými cenami [4].

1.2 Popis

Hru může hrát 2 až 6 hráčů, kteří se střídají v pokládání kartiček na hrací desku. Hrací deska má velikost 8 x 8 políček a je dokola ohraničena stanicemi metra. Další čtyři stanice se nacházejí uprostřed hracího pole. Úkolem hráčů je postavit co nejdelší tratě vedoucí z jejich stanic. K tomu slouží hrací kartičky, na kterých jsou zobrazeny kolejnice. K dispozici je 24 druhů kartiček, které se liší směry svých kolejnic. Pokládáním kartiček na hrací desku tak vznikají různě dlouhé a klikaté tratě. Kartičky jsou čtvercového tvaru a na každou její stranu je napojen jeden vstup kolejnice a jeden výstup. Jedna kartička tedy může být součástí až čtyř různých tratí.

1.3 Průběh hry

Před začátkem hry jsou stanice při okraji hrací desky přiděleny jednotlivým hráčům tak, aby všichni hráči měli stejný počet stanic a aby byly spravedlivě rozmístěné. Okrajových stanic je 32, takže při určitém počtu hráčů zůstanou některé stanice nepřiděleny. Rozdělení stanic je pevně dané pro každý počet hráčů.

Dále se zamíchají kartičky a položí se lícem dolů. Každý hráč si vytáhne náhodně jednu kartičku a s tou bude hrát.

Hráči se postupně střídají v tazích. Hráč, který je na tahu musí umístit svojí kartičku na hrací desku, při tom musí dbát na následující pravidla:

- kartička musí být položena na prázdné políčko
- kartička se musí alespoň jednou stranou dotýkat jiné, již položené kartičky, nebo okraje hrací desky
- orientace kartičky je pevně daná, není povoleno kartičky otáčet
- položením kartičky nesmí vzniknout trasa dlouhá pouze jedno políčko (toto neplatí, pokud jinak nelze kartičku položit)

Po položení kartičky si hráč vytáhne další kartičku a na tahu je další hráč. Hra končí ve chvíli, kdy jsou všechny kartičky umístěny na hrací desce [5].

1.4 Bodování

Ihned po propojení trati se stanicí se hráči přičtou za tuto trať body. Za každý průjezd kartičkou se přičítá jeden bod. Prochází-li tedy trať jednou kartičkou vícekrát, přičte se tolik bodů, kolikrát se přes kartičku projíždí. Pokud končí trať v některé ze středových stanic, počet bodů se navíc zdvojnásobí.

Vítězí hráč, který má po skončení hry na kontě nejvíce bodů.

1.5 Strategie

Hráči se každým tahem zpravidla snaží co nejvíce prodloužit svoje rozestavěné tratě a pokud možno je navíc přiblížit středovým stanicím. Často je potřeba volit mezi těmito veličinami – mám radši prodloužit tuto trasu o více políček s tím, že se vůbec nepřiblížím, nebo naopak vzdálím středu pole? Nebo prodloužit jinou trasu, byť jen o jedno pole, ale za to zvýšit pravděpodobnost, že se počet bodů bude dvojnásobit?

Ale prodloužit vlastní trasu není jediným možným taktickým manévrem. Mnohdy je výhodnější naopak zatarasit cestu soupeři a nedovolit mu rozvinout jeho trať hlouběji do pole. Zvláště pokud je jeho trasa už značně dlouhá a hrozí reálné nebezpečí dvojnásobení bodů za dosažení středové stanice. V tom případě je snahou poslat soupeřovu trasu nejkratší cestou ke kraji.

Tyto a mnohé další okolnosti je třeba zvážit při každém tahu. I sebedůkladnější analýzu situace však může zhatit ještě jeden prvek – náhoda. Naše rozhodování je nejvíce závislé na tom, kterou kartičku si zrovna vytáhneme. Těžko můžeme kalkulovat, že když v tomto tahu umístíme kartičku semhle a v příštím jí umístíme támhle, tak soupeř nemá šanci. Narozdíl od her, jako jsou šachy, není možné deterministicky odvodit nejlepší strategii. Možná až v konečných fázích hry, kdy zbývá několik posledních kartiček, lze z hrací desky odvodit, které kartičky ještě chybí, a podle toho promyslet všechny možné varianty. Na tom ovšem nelze postavit celou herní strategii.

2 Monte Carlo Tree Search

Monte Carlo Tree Search je vyhledávací metoda typu „best-first“, jejíž výhodou je, že není založena na ohodnocování uzlů vyhledávacího stromu heuristickou funkcí. Metoda buduje svůj vyhledávací strom na základě výsledků náhodných herních simulací. Díky vysokému počtu simulací pak může lépe určit slibné tahy. Metoda může být použita na kteroukoliv konečnou hru, která splňuje alespoň tyto tři podmínky:

- výsledky simulací mají omezený obor hodnot
- úplná informace (s neúplnou informací nemají simulace v základní metodě smysl)
- simulace končí v rozumném čase (hra je konečná)

Ve vyhledávacím stromu každý uzel reprezentuje jeden herní stav. Uzel musí obsahovat alespoň tyto dvě informace:

- hodnota uzlu (většinou součet nebo průměr výsledků simulovaných her, které prošli tímto uzlem),
- počet průchodů uzlem.

Na začátku algoritmu obsahuje strom pouze kořen, jež reprezentuje aktuální stav hry. Pak buduje strom ve čtyřech krocích:

1. **Selekce** Prochází strom od kořene, dokud nedojde do listu. Vybere další herní stav, který není součástí stromu.
2. **Expanze** Přidá vybraný stav do stromu jako nový uzel.
3. **Simulace** Z dosaženého herního stavu pomocí náhodných tahů všech hráčů dohraje hru do konce.
4. **Backpropagation** Výsledek simulované hry je promítnut do všech uzlů stromu, kterými se prošlo.

Tato sekvence kroků se neustále dokola opakuje, dokud má algoritmus čas. Nakonec se vybere tah, který odpovídá nejlepšímu potomku kořene stromu.

2.1 Strategie v jednotlivých krocích

2.1.1 Selekce

Při procházení stromu od kořene se následovník aktuálního uzlu vybírá podle určité strategie. Tato strategie by měla udržovat rovnováhu mezi explorací (exploration) a exploatací (exploitation). Samozřejmě je lepší častěji vybírat tahy, které vypadají slibně a mají šanci na dobrý výsledek (exploatace). Na druhou stranu je nutné procházet i tahy, které byly dosud prozkoumány nedostatečně a jejich hodnota může být podceněna(explorace).

Modelový příklad exploration-exploitation dilematu je problém Multi-Armed Bandit. Při selekci lze na každý uzel stromu společně s jeho potomky nahlížet jako na samostatnou instanci MAB. Každý z potomků uzlu představuje jedno rameno automatu s neznámou výplatní funkcí. Úkolem je vybrat jednoho potomka na základě znalosti výsledků předchozích voleb, při čemž se snažíme o celkově nejlepší výsledek. Celý průchod stromem během selekce tak můžeme brát jako posloupnost na sobě nezávislých instancí problémů MAB s počátkem v kořenu stromu.

K výběru potomků při průchodu stromem se používá algoritmus UCT neboli Upper Confidence Bounds for Trees. Ten v jednotlivých uzlech využívá metodu UCB1.

UCB1 v první fázi vybere každé rameno jednou. V druhé fázi vybirá rameno, které maximalizuje vzorec

$$X_i + C\sqrt{\frac{2\log n}{n_i}}$$, kde

- X_i je průměrná hodnota ramena i ,
- n je celkový počet průchodů vrcholem,
- n_i je počet her, v kterých bylo vybráno rameno i .

První sčítanec X_i zajišťuje exploitaci, zatímco druhý sčítanec exploraci. Konstanta C umožňuje parametrizovat poměr explorace a exploitace.

2.1.2 Expanze

Expanze přidává nové uzly do stávajícího stromu. Zpravidla není možné udržovat celý herní strom v paměti, proto expanzní strategie určuje, zda daný uzel rozšířit o jednoho či více potomků. Rozvinutý strom chceme mít celý v paměti. Běžně se používá strategie přidání pouze jednoho potomka, který je určen již při selekci. Tato strategie je jednoduchá a časově i paměťově nenáročná. Lze ale použít i jiné strategie. Například je možné expandovat strom do určité hloubky (např. 2, 3) ještě před zahájením vyhledávání. Nebo je možné přidat do stromu všechny potomky daného uzlu. Zde už je ale nutné počítat s větší paměťovou náročností. Dopad expanzní strategie na výslednou efektivitu algoritmu minimální. Je-li potřeba, dá se správnou volbou této strategie ušetřit jisté množství paměti, nicméně podstatný vliv na sílu hráče to nemá.

2.1.3 Simulace

Simulace znamená dohrání hry do konce, v původní variantě pomocí náhodně zvolených tahů. Použití čistě náhodných tahů ovšem není ideálním řešením, taková strategie není příliš silná. Je proto lepší použít simulační strategii k výběru pseudonáhodných tahů. Cílem je vybírat v simulaci lepší tahy pomocí nějaké heuristiky. Zvolená strategie má na výslednou sílu hráče významný vliv.

Při volbě simulační strategie je důležité vzít v úvahu dva aspekty. Prvním je složitost použité heuristiky. Čím přesněji ohodnotíme každý tah, tím budou výsledky simulací spolehlivější, budou pro nás mít větší význam. Na druhou stranu pokud bude heuristika příliš časově náročná, počet simulací, které se stihnou provést za časovou jednotku, rapidně klesne. Tím klesne i konečná síla hráče. Druhým aspektem je míra náhody, která hraje roli při volbě tahů v simulaci. Pokud budou tahy vybírány příliš náhodně, slabé tahy budou hrány s větší pravděpodobností, význam heuristiky slábne a s ním i síla strategie. Naopak s nedostatečnou mírou náhody nastává druhý extrém. Pro daný herní stav se bude často opakovat stejný tah, průběh simulací z daného stavu bude jednotvárný a síla strategie tím opět klesá.

Sestavení efektivní simulační strategie je velice obtížné, nicméně je to jedna z klíčových částí pro vytvoření kvalitního MCTS hráče.

2.1.4 Backpropagation

Backpropagation je krok, ve kterém se výsledek simulované hry zpětně promítne do těch uzlů, které algoritmus navštívil při průchodu od kořene do listu. Obor hodnot výsledku závisí na typu hry a na konkrétní implementaci. U hry pro dva hráče s nulovým součtem stačí rozlišovat 3 možné výsledky: výhra, remíza, prohra (+1, 0, -1).

V tomto kroku zvolená strategie určuje, jakým způsobem se bude aktualizovat hodnota *v* uzlu.

- **Průměr**. Nejpoužívanější je strategie průměru, která jednoduše do rodičovského uzlu předá aritmetický průměr výsledků simulovaných her, které tímto uzlem prošly. Tuto strategii poprvé použili Kocsis a Szepesvári [2] a od té doby je považována za nejefektivnější a nejvíce se používá.

V literatuře jsou navrhovány i další možné „backpropagační“ strategie, zpravidla jde pouze o slabší strategie. Žádná strategie zatím neprokázala vyšší efektivitu než strategie průměru [1].

2.2 Výběr výsledného tahu

Po ukončení simulací je na čase vybrat výsledný tah, který se provede ve skutečné hře. Tento tah odpovídá „nejlepšímu“ potomku kořene MCTS stromu. Existuje opět několik metod, jak určit, který potomek je nejlepší.

- Nejvyšší hodnota
- Nejvyšší počet návštěv
- Nejvyšší hodnota i počet návštěv. Pokud neexistuje potomek, který má zároveň nejvyšší hodnotu i nejvyšší počet návštěv, provádějí se další simulace, dokud se takový potomek neobjeví.

3 MCTS ve hře Metro

Jelikož Metro je hra poměrně nová a ne moc známá, oblast umělé inteligence v této hře není dostatečně prozkoumána. Na rozdíl od her jako jsou šachy nebo GO, pro Metro neexistují taková kvanta výzkumných článků studujících efektivitu jednotlivých algoritmů a metod pro umělou inteligenci. Neexistuje žádná metoda, která by byla obecně známá jako nejlepší pro tuto hru. Nemáme tedy s čím porovnávat.

Hloubka stromu

Výhodou hry Metro pro použití MCTS je konečnost. Každým tahem přibyde na hrací desku jedna hrací karta a hra končí v momentě, kdy je hrací deska zaplněná. To znamená, že každá hra trvá 60 tahů To je samozřejmě příznivé pro algoritmus MCTS, hloubka vyhledávacího stromu bude vždy nejvíce 60.

Faktor větvení

Podobné omezení platí i u faktoru větvení stromu. Kdyby bylo povoleno pokládat karty na kterékoliv místo na desce, maximum počtu možných tahů v každém tahu by bylo 60. Díky pravidlu, podle kterého lze táhnout jen na místa, ke kterým již vede nějaká trať od některé stanice na kraji hrací desky, je toto číslo nižší. Pokusíme se toto číslo omezit co nejpřesněji. Při prvním tahu může hráč položit svojí kartu pouze k okraji hrací desky, to je maximálně 28 možností. Položením karty na nové místo se někdy počet dostupných pozic zvýší, v nejlepším případě o 2: jeden soused bude vždy obsazený, mohli tedy vzniknout 3 nově hratelná políčka, ale jednu volnou pozici jsme položením kartičky obsadili. Uvažujme nyní, že položením každé kartičky přibydou 2 hratelná pole. Po prvních 10 tazích máme 48 hratelných, 50 volných, dalším tahem by zbylo už jen 49 volných a dále už by toto číslo jen klesalo. Prakticky se mi podařilo dosáhnout 46 hratelných pozic. Faktor větvení tak můžeme shora ostře omezit číslem 50. Pomocí exmperimentů jsme zjistili, že průměrná hodnota faktoru větvení je přibližně 18.

3.1 Naše implementace

V této sekci si rozebereme naší implementaci MCTS pro hru Metro. Podíváme se, jaké strategie jsme zvolili pro jednotlivé kroky metody MCTS a proč.

3.1.1 Strategie selekce

Pro selekci jsme použili metodu UTC. Přes její jednoduchost na implementaci je velice účinná a pro naše účely dostačující. Pro každého potomka i spočítáme hodnotu

$$UCT_i = \frac{n_i}{v_i + \varepsilon} + \sqrt{\frac{\ln(n_u + 1)}{n_i + \varepsilon}} + r * \varepsilon \quad ,$$

kde

- n_i je dosavadní počet průchodů potomka i ,
- v_i je celková hodnota potomka, nu je počet průchodů aktuálního uzlu,
- r je náhodné číslo, které rozliší případné shodné výsledky.

Potomek, který bude mít tuto hodnotu nejvyšší, bude vybrán.

3.1.2 Strategie expanze

Expanze má za úkol přidat do stromu nový uzel vybraný při selekci, případně i více uzlů. Naše expanze přidává kromě vybraného uzlu ve stejném kroku také všechny jeho sourozence, kteří se ve stromě vyskytnou. funguje tak, že v momentě, kdy selekce dojde do uzlu bez potomků, zvolí se příslušný počet potomků a ti všichni jsou do stromu přidáni. Následně se provede selekce ještě jedenkrát a tím je vybrán nový uzel, z kterého odstartuje simulace.

Jak jsme uvedli výše, ve hře Metro je v každém okamžiku maximálně 50 možných tahů. Některé tahy jsou ale v daný moment na první pohled špatné a je zbytečné s nimi kalkulovat. V našem algoritmu by jimi proběhlo množství simulací, které by bylo užitečnější u uzlů, jež mají šanci na úspěch.

K výběru „vyvolených“ potomků využíváme pomocnou ohodnocovací funkci, kterou popíšeme dále. Touto funkcí ohodnotíme všechny možné potomky daného uzlu, ale do stromu přidáme jen ty nejlepší. Tím se zbavíme nesmyslných tahů, ušetříme místo v paměti i čas.

Jaký počet potomků do stromu přidat určíme parametrem. Hodnoty tohoto parametru budem studovat v kapitole Experimenty.

Použitá heuristika

Měli jsme za úkol vytvořit heuristiku, která nám pomůže odhadnout "kvalitu" tahů, které může hráč zahrát. Heuristika byla konstruována s ohledem na jednoduchost. Důmyslnější heuristika by poskytovala přesnější odhady tahů, nicméně její přílišná časová náročnost by měla negativní vliv na výslednou efektivitu algoritmu.

Hledáli jsme funkci, která pro danou hrací kartu a dané pole na hrací ploše vrátí hodnotu z intervalu (min, max). čím výhodnější tah, tím vyšší hodnota.

K ohodnocení tahu jsme dvě veličiny:

- prodloužení stávajících tras
- přiblížení trasy ke středu

Čím více tah prodlouží některou z hráčových tras, tím je pro něj výhodnější. Na druhou stranu tah, který prodlouží hráči trasu např. o 2 jednotky, ale zároveň jí přiblíží těsně k okraji, bývá horší než tah, který trasu prodlouží pouze o 1 jednotku, ale přiblíží jí ke středu a tím zvýší pravděpodobnost na zdvojnásobení bodového ohodnocení trasy.

Ohodnotíme každou kolejnici na kartičce a jednotlivé hodnoty jednoduše sečteme. Hodnoty ohocnocení kolejnic spadají do intervalu $\langle -1\,;1\rangle$. Záporné hodnoty patří kolejnicím, které prodlouží trasu některého ze soupeřů. Hodnotu celého tahu dostaneme součtem hodnot všech 4 kolejnic. Výsledný interval hodnot je $\langle -4,4\rangle$

3.1.3 Strategie simulace

Simulace jsou základním stavebním kamenem metod MCTS. Kvalita simulační strategie má velký vliv na celkovou úroveň MCTS hráče. Klíčem je nevybírat v simulaci skutečně náhodné tahy, ale pokusit se najít lépe vypadající tahy, které přidají simulacím na realističnosti.

Naše simulační strategie používá stejnou heuristickou funkci, kterou používáme v kroku expanze. V každém uzlu ohodnotíme všechny možné tahy ohodnocovací funkcí (viz výše) a vybereme z nich daný počet těch nejlepších. Těmto pak přiřadíme pravděpodobnost jejich zvolení. Čím vyšší hodnocení, tím vyšší pravděpodobnost, že bude tah v simulaci zvolen. Nakonec se na základě přiřazených pravděpodobností vybere náhodně jeden tah a ten bude simulován.

3.1.4 Strategie backpropagation

Volba strategie v kroku backpropagation byla jasná. Jak bylo zmíněno v kapitole 2.1.4, v podstatě jediná používaná je strategie průměru.

Pro hru Metro je navíc potřeba vymyslet funkci, která ze skóre hráčů dosaženého v simulované hře vytvoří jedinou hodnotu, která bude propagována zpět do uzlů. Nadefinujeme si obor hodnot této funkce jako $\langle 0,1\rangle$. Kdyby se jednalo o hru dvou hráčů, ohodnocení vítěze by bylo blízko číslu 1, prohrávající by se blížil k 0. Jak blízko k těmto hranicím by záleželo na získaných bodech obou hráčů. Vítězství 100:20 je zjevně lepší než např. vítězství 70:68 a podle toho by také mělo být ohodnoceno. Metro ale může hrát až 6 hráčů, jakou funkci tedy použít zde? Nic se nemění na tom, že hodnota vítězství se bude blížit 1. Otázkou je, jak ohodnotit ostatní umístění.

Naše funkce odráží myšlenku, že skončit druhý je jen o málo lepší než skončit poslední, obojí je prohra. Funkce je nastavitelná pomocí dvou parametrů, α a β. Parametrem α nastavíme, jak velký podíl na výsledné hodnotě má vítěztví ve hře. Jinak řečeno, výsledek hry bude ohodnocen číslem z intervalu $\langle \alpha , 1\rangle$ Parametr β pak určí interval, z kterého budou ohodnocováni všichni poražení. Tento interval se rozdělí rovnoměrně podle počtu hráčů. Např. při hře 4 hráčů budou ohodnocení výsledku hry vybírána podle umístění z těchto intervalů:

- 1. místo: $(\alpha, 1)$
- 2. místo: $(\frac{2}{3}\beta, \beta)$
- 3. místo: $(\frac{1}{3}\beta, \frac{2}{3}\beta)$
- 4. místo: $(0, \frac{1}{3}\beta)$

Hodnotami parametrů α i β se budeme zabývat v sekci Experimenty.

4 Experimenty a výsledky

4.1 Způsoby testování

Herní engine

K testování jsme využili náš herní engine. Pro přímé porovnání dvou různých hodnot sledovaného parametru nám postačila hra dvou hráčů. Množství simulací pro jeden tah jsme nastavili na 1000, každý experiment jsme testovali na vzorku cca 100 her.

4.2 Experimenty

4.2.1 Použití heuristiky

Naší heuristiku jsme testovali v závislosti na velikosti větvícího faktoru a počtu simulací. U hry dvou hráčů měl žlutý hráč heuristiku k dispozici, modrý hráč nikoliv. Ukázalo se, že naše heuristika je platná pouze při omezených podmínkách.

Pokud omezíme větvící faktor v simulacích, dává lepší výsledky hráč používající naši heuristiku. Pokud můžeme při simulacích v každém kroku pracovat s omezeným počtem možných tahů, je zjevně výhodnější vybrat slibné tahy určitou heuristikou než pracovat s čistě náhodnými tahy. Naopak pro neomezený větvící faktor se naše heuristika neosvědčila, žlutý hráč drtivou většinu her prohrál. Příčina je patrně ta, že v naší heuristice jsme vyřadili tahy, které se nám zdály na první pohled špatné. Například tah, který prodlužuje pouze soupeřovy trasy. Takovým tahům jsme nedali vůbec šanci a to se zřejmě nevyplatilo. Výsledky testu jsou zaznamenány na obrázku 3.1.

Dalším faktorem, kde by se naše heuristika mohla uplatnit, je počet simulací. Nízký počet náhodných simulací nemusí prozkoumat dostatečně všechny tahy, a proto by použití heuristiky mohlo pomoct. Dle experimentů se tato úvaha potvrdila, nicméně zlepšení není tolik výrazné. Pro tento test jsme použili neomezený větvící faktor. Když jsme nastavili počet simulací za jeden tah na 100, poměr výher žlutého hráče byl pouze okolo 53%, při 50 simulacích za tah už to bylo více než 60%.

Výsledky jsou zaznamenány na obrázku 3.2.

4.2.2 Výsledková funkce

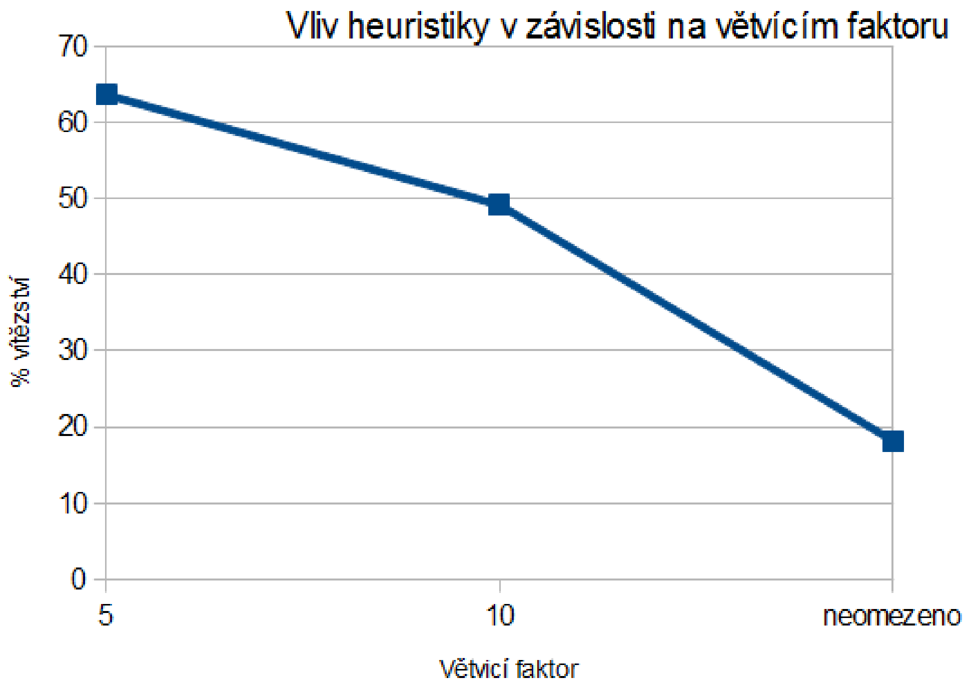
Další experiment se týkal parametrů α a β použitých ve funkci, která ohodnotí výsledek nasimulované hry. Po dokončení každé simulace je potřeba určit, jak dobrá tato simulace pro hráče je. Pro nás je nejdůležitější celkové umístění. Hodnocení simulace v případě výhry by mělo být značně vyšší, než hodnocení pro prohraného hráče. Určili jsme, že ohodnocení bude spadat do intervalu (0;1).

Parametr α určuje horní část intervalu, z které bude pocházet hodnocení pro vítězství ve hře. Před začátkem experimentu jsme se domnívali, že ideální hodnota parametru α bude pravděpodobně velice blízko horní mezi intervalu, tedy číslu 1. Zvolili jsme tedy hodnotu 0,96 jako výchozí hodnotu pro náš experiment a zajímalo nás, jak velký vliv má tato hodnota na výslednou sílu našeho počítačového hráče.

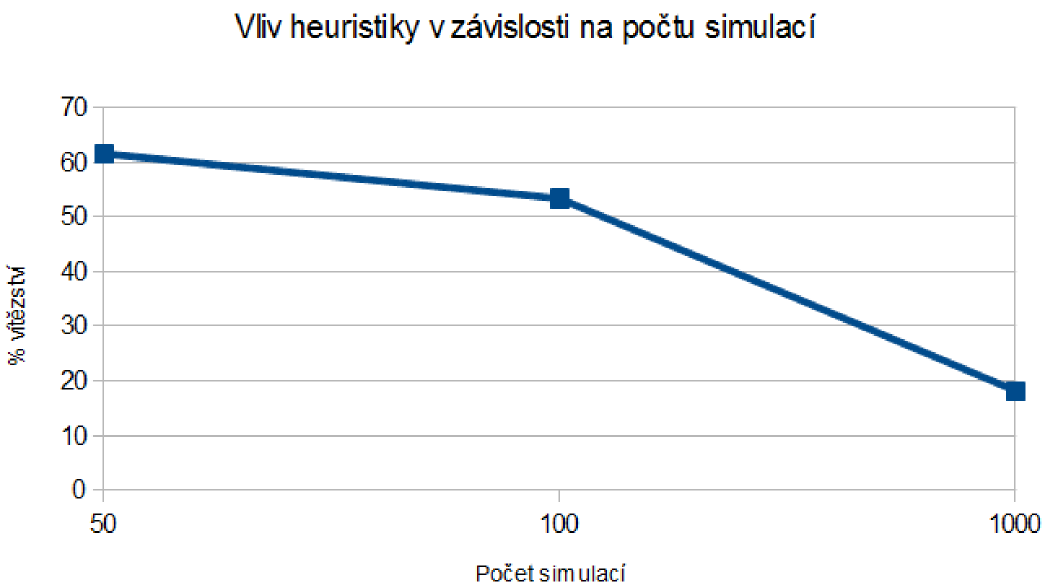
Výsledky experimentů prokázaly, že parametr α je důležitý. Potvrdilo se, že pro nejlepší výsledky hráče by tento parametr měl být nastaven velice blízko číslu 1. Například oproti hodnotě 0,5 zvítězila naše zvolená hodnota 0,96 v 68% her, což jistě není zanedbatelný poměr. Výsledky testu jsou vidět na obrázku 3.3.

Parametr β má opačný význam než parametr α. Určuje spodní část intervalu (0;1), do které patří ohodnocení her v prohrávajících případech. Parametr β by měl být dostatečně vzdálen od parametru α. Jelikož parametr α nám v experimentech vyšel nejlépe, když byl nastaven blízko k číslu 1, určili jsme jako výchozí hodnotu 0,5 parametru beta.

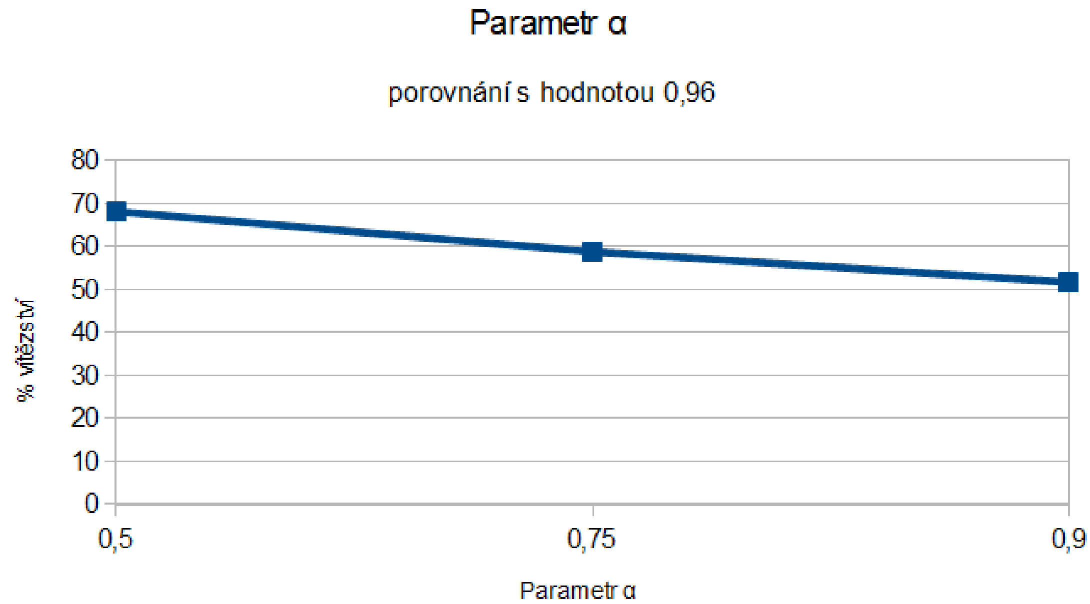
Experimenty neprokázaly vliv parametru β na výslednou sílu počítačového hráče.



Obrázek 3.1: Efektivita heuristiky podle větvícího faktoru



Obrázek 3.2: Efektivita heuristika podle počtu simulací



Obrázek 3.3: Testování parametru α

Závěr

V této bakalářské práci jsme popsali metodu Monte Carlo Tree Search a možnosti její implementace. MCTS je vyhledávací metoda založená na simulacích, která v posledních letech zažívá vzestup v oblasti teorie her. U spousty her je největším problémem navržení vhodné funkce pro ohodnocení herních stavů. MCTS se díky tomu, že ohodnocovací funkci nepotřebuje, stává vhodným adeptem pro tyto hry.

Dále jsme si představili deskovou hru Metro. Její to hra s úplnou informací a je konečná, což jsou dobré předpoklady k použití MCTS. Popsali jsme vlastní implementaci MCTS pro hru Metro. Dále jsme navrhli jsme vlastní heuristiku pro použití v herních simulacích MCTS a sledovali jsme zlepšení úrovně počítačového hráče, které tato heuristika měla přinést. Pomocí experimentů jsme zjistili, že naše heuristika zvýší kvalitu počítačového hráče, pouze když omezíme větvící faktor nebo počet provedených simulací pro jeden tah. Ukázalo se, že pokud neomezíme větvící faktor vyhledávacího stromu a necháme algoritmus provést pro každý tah dostatečný počet simulací, jsou náhodné simulace silnější než simulace s naší heuristikou.

Součástí práce je také počítačový program Metro, což je herní prostředí pro 2 až 6 hráčů. Hráči mohou být ovládání jak počítačem tak člověkem, takže čtenář si může hru vyzkoušet. K dispozici jsou 3 úrovně počítačového hráče. Nejtěžší varianta je skutečnou výzvou, mně osobně se jí nepodařilo porazit.

Seznam použité literatury

[1] G. Chaslot: Monte Carlo Tree Search, Disertační práce, Maastricht University, https://project.dke.maastrichtuniversity.nl/games/files/phd/Chaslot_thesis.pdf, 2010

[2] L. Kocsis & C. Szepesvari: Bandit based monte-carlo planning. In *In:*

ECML-06. Number 4212 in LNCS, pages 282–293. Springer, 2006.
[3] S. Russell, P. Norvig: Aritificial Intelligence, A Modern Approach, *Prentice Hall*, Englewood Cliffs, USA, 2003

[4] Interview with Dirk Henn, http://www.brettspillguiden.no/begeret/?nid=3

[5] Wikipedia, Metro(hra), http://cs.wikipedia.org/wiki/Metro_(hra)

[6] C. Browne, E. Powley, D. Whitehouse a kol.: A Survey of Monte Carlo Tree Search Methods, IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, vol. 4, no. 1, 2012

[7] T. Kozelek: Methods of MCTS and the game Arimaa, Master's Thesis, MFF UK, Praha, 2009