

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentácia k projektu

(Strojové učení a umělá inteligencia)

Dátum: 5. januára 2020

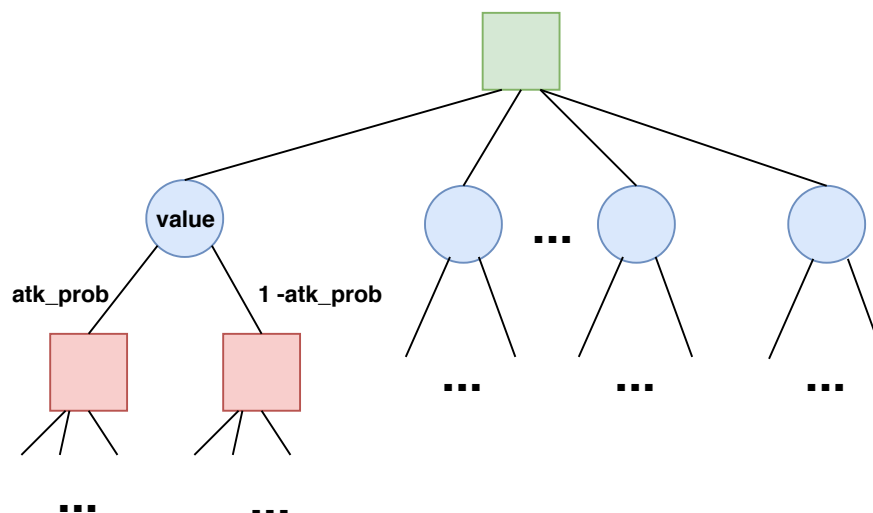
Vladimír Marcin, (xmarci10)
Žofia Vrábl'ová, (xvrabl03)
Tomáš Zubrik, (xzubri00)

1 Expectimax

1.1 Hlavná myšlienka

Na určenie najlepšieho ťahu sme použili algoritmus Expectimax, ktorého výsledkom je ťah, ktorý maximalizuje úžitok hráča.

Na začiatku ťahu nášho hráča uvažujeme všetky možné útoky, z ktorých je následne zvolený ten najlepší. Pre každý z týchto možných útokov musíme zvážiť 2 situácie: *útok sa podaril* – získanie políčka, na ktoré útočíme, alebo *útok sa nepodaril* – strata kociek z útočiaceho políčka (až na 1). Táto situácia je znázornená na obrázku 1.



Obr. 1: Zelené políčko predstavuje aktuálny stav hry, pričom na ťahu je naša AI. Modré políčka predstavujú všetky možné útoky, pričom každý z nich môže skončiť úspechom (s pravdepodobnosťou atk_prob) alebo neúspechom (s pravdepodobnosťou $1-\text{atk_prob}$). Následne je na ťahu protihráč (červené políčko).

Na to, aby sme mohli nejakým spôsobom ohodnotiť všetky možné útoky (modré políčka na obrázku 1), musíme zobrať do úvahy všetky možné ťahy našich protihráčov, pričom v konečnom dôsledku zvolíme taký ťah, kde je hodnota (**value**) najvyššia.

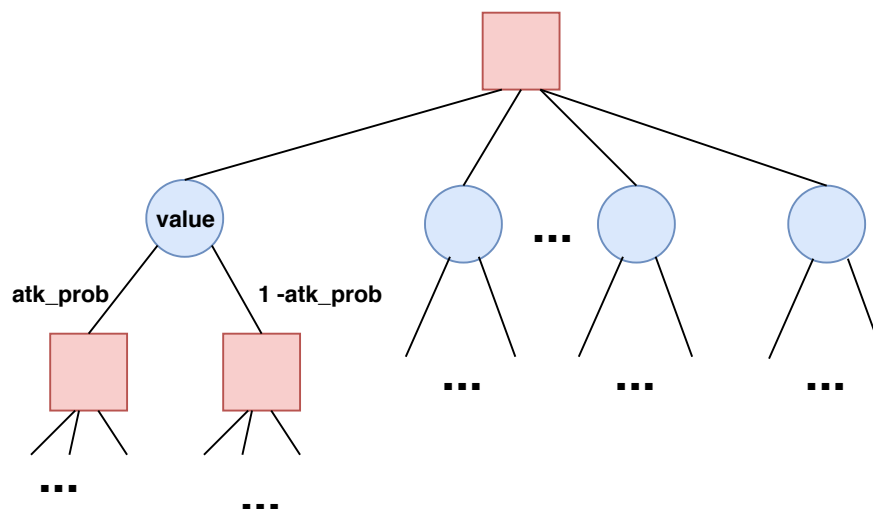
Na každej úrovni stromu, kde je na ťahu protihráč, potrebujeme zvážiť všetky jeho možné útoky (Obr. 2) a vybrať práve ten, ktorý maximalizuje jeho úžitok (čo predstavuje snahu protihráča zvíťaziť). Takýchto úrovní je pre každého protihráča niekoľko, keďže v jednom kole má k dispozícii viacero ťahov. To spôsobuje exponenciálny nárast počtu stavov, čo znemožňuje preskúmať všetky možné ťahy v danom časovom limite. Kvôli tomu sme zvolili alternatívny prístup vysvetlený ďalej v texte.

2 Funkcionalita AI agenta

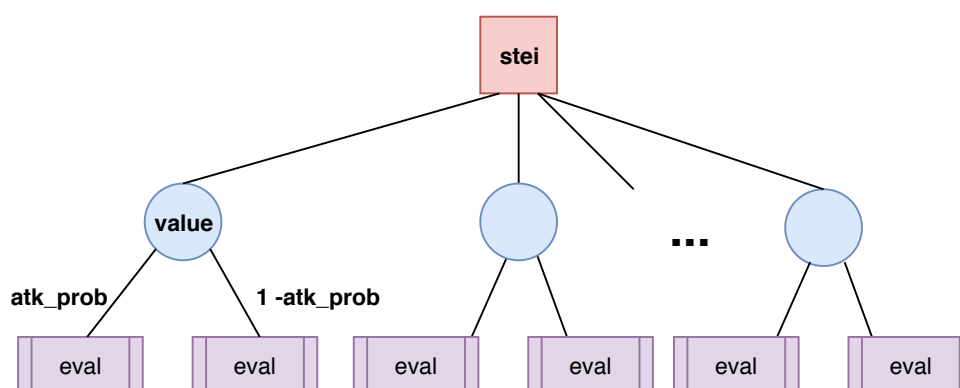
Pri návrhu AI agenta sme vychádzali z vyššie spomenutých problémov a snažili sme sa zmenšiť prehľadávaný stavový priestor. Jednou z použitých metód je prerezávanie, ktorej sa bližšie venuje sekcia 2.1.1.

Druhým prístupom je odstránenie nutnosti prehľadávania všetkých možných stavov každého súpera. Preto na určenie "najlepšieho" ťahu nášho súpera zavoláme metódu `ai_turn` z modulu STEI (ktorý bol zvolený pre vykazovanie najlepších výsledkov pri testovaní).

STEI¹ je AI agent, ktorý predstavuje Single Turn Expectminimax. Základná myšlienka je znázornená na obrázku 3.



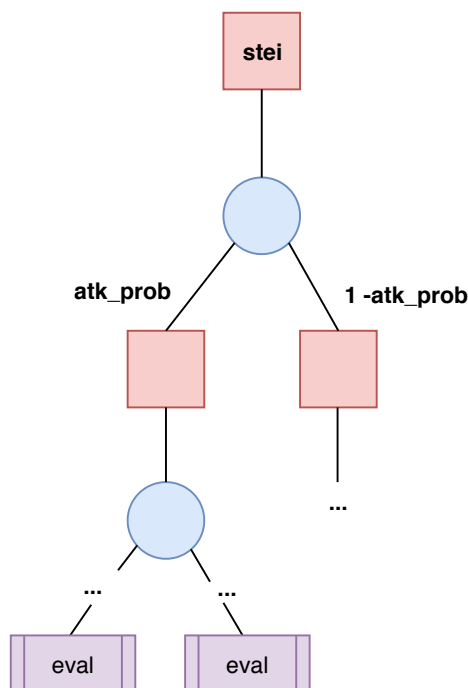
Obr. 2: Koreň herného stromu na obrázku predstavuje aktuálny stav, pričom na ťahu je protihráč. Protihráč chce zvoliť svoj najlepší ťah berúc do úvahy svoje nasledujúce ťahy.



Obr. 3: STEI agent uvažuje všetky ťahy (modré políčka), z ktorých na základe pravdepodobnosti úspešného útoku a pravdepodobnosti udržania získaného políčka do nasledujúceho ťahu vyberie najlepší možný ťah.

STEI agent postupne vyhodnocuje všetky svoje ťahy, až kým metóda `ai_turn` nevráti `EndTurnCommand` (Obr.4). Ak funkcia `aiturn` vráti `EndTrunCommand`, predá riadenie ďalšiemu protihráčovi alebo ak je posledný, na listovej úrovni sa zavolá funkcia `eval`. Priebeh hry viacerých hráčov je znázornený na obrázku 5.

¹https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=180901



Obr. 4: V koreni stromu je zavolaná funkcia `aiturn`, ktorá vráti najlepší možný ťah (modré políčko). Následne je dvakrát volaná funkcia `aiturn` s novým stavom hry (útok bol úspešný / neúspešný).

2.1 Pruning

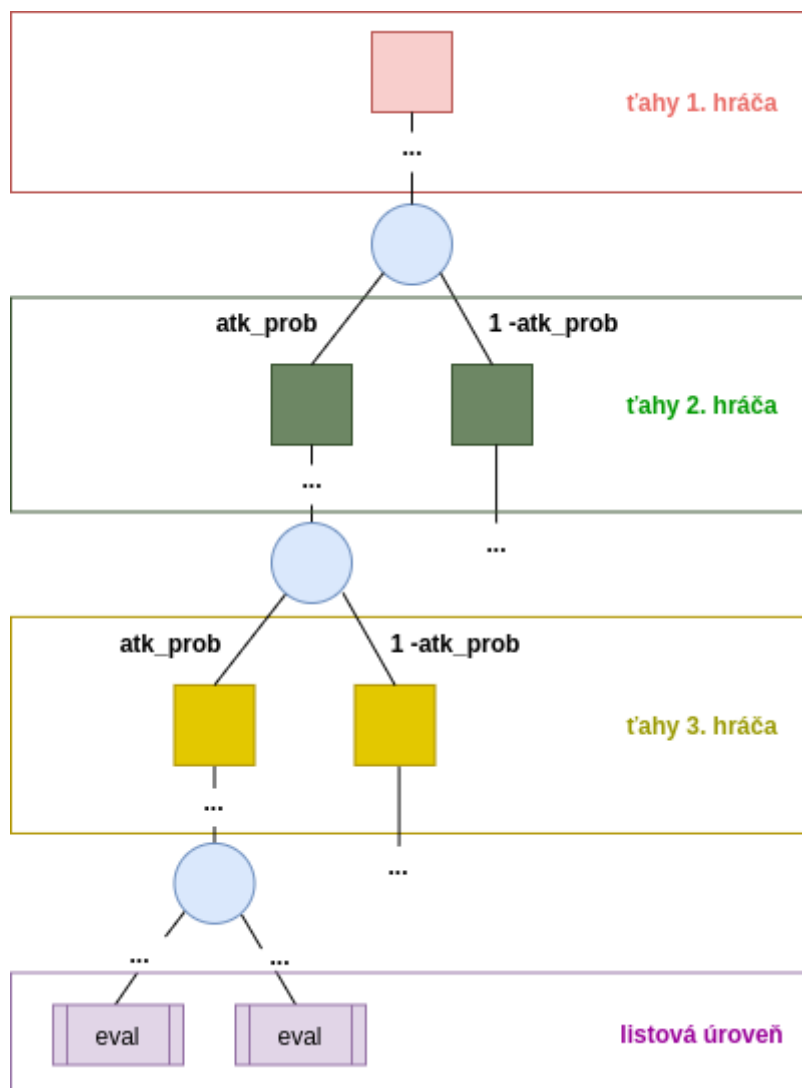
2.1.1 Forward pruning

Dopredným prerezávaním stavového priestoru zabezpečíme, že sa bude v konečnom dôsledku vyhodnocovať menej stavov, čím sa výrazne urýchli výpočet. Odrezané sú všetky uzly, v ktorých je pravdepodobnosť získania políčka menšia ako 0.47, čo je hodnota pravdepodobnosti útoku 8 kociek na 8 súperových kociek.

Tento prístup predstavuje riziko, kvôli tomu, že môže nastať situácia kedy odrežeme najlepší ťah. Náš prezentovaný prístup je podobný konaniu človeka, pretože človek tiež neuvažuje na prvý pohľad nevýhodné ťahy.

2.1.2 Cutoff test

Ak je pravdepodobnosť skúmaného útoku (`atk_prob`) vyššia ako 0.8, zanorovať sa bude iba vetva predstavujúca úspešný útok. Pre druhú vetvu bude hneď vyhodnotená funkcia `eval`.



Obr. 5: Herný strom znázorňujúci priebeh hry 3 protihráčov.

3 Evaluačná funkcia

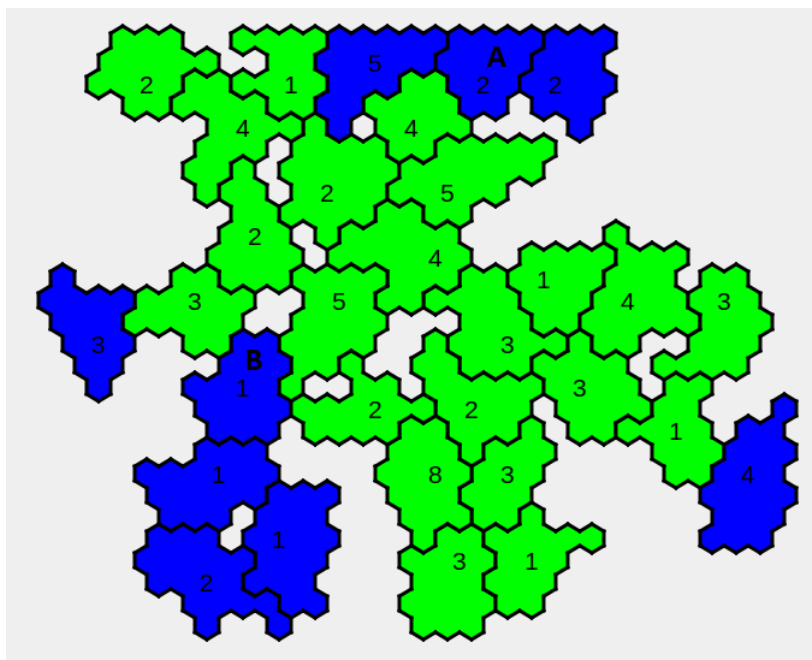
Pri vytváraní evaluačnej funkcie sme kládli hlavný dôraz na to, aby najlepšie ohodnocovala stavy hry s najvyšším skóre, a tým nútila agenta vyberať ťahy, ktoré vedú k získaniu čo najväčšej spojitaj oblasti hernej plochy.

Tento predpoklad sme rozšírili o ďalšie dva potenciálne zaujímavé atribúty vedúce k lepším výsledkom. Prvým z nich je počet hraničných oblastí s počtom kociek rovným 1. Tieto oblasti sú ľahkým terčom pre útoky protihráčov a navyše z nich nie je možné rozširovať získané územie, pokiaľ nebudú pridané ďalšie kocky po ukončení ťahu. Je žiadúce, aby hráč vlastnil čo najmenej takýchto hraničných oblastí. Na obrázku 6 je ich počet pre modrého hráča 1.

Druhým spomínaným atribútom sú hraničné oblasti, ktoré sú ohrozené protihráčmi. Za ohrozenú oblasť pritom považujeme takú oblasť, v ktorej susedných oblastiach obsadených protihráčmi sa nachádza vyšší počet kociek. Avšak, je potrebné rozlišovať celkový rozdiel počtu kociek, nielen počet hraničných oblastí, ktoré sú týmto spôsobom ohrozené.

Uvažujme situáciu, kedy máme vybrať spomedzi dvoch ťahov. Ťah č. 1 vedie k získa-

Na obrázku 6 je uvedený príklad hracej plochy, v ktorej má modrý hráč 2 ohrozené hraničné oblasti (označené A, B). Celkový súčet prevyšujúcich kociek je rovný 9, čo je číslo vstupujúce do evaluačnej funkcie.



Evaluačná funkcia musí určovať kladné ohodnotenia uzlov, preto je definovaná nasledovne:

kde *board* je hracia plocha, *s* je skóre, *abd* (attacking boarder dices) je celkový počet ohrožujúcich kociek súpera/súperov v hraničných oblastiach, *naa* (not attacking areas) je počet hraničných oblastí s počtom kociek 1.

V prahovom prípade, ak by nejaký ťah hry viedol k nulovému skóre, hodnota evaluačnej funkcie bude nulová, čím ostane splnená podmienka kladného ohodnotenia stavov.

4 Testovanie a výsledky

Počas formovania správania agenta prebiehalo testovanie súbežne s jeho vývojom, pričom sme vždy najskôr zisťovali správnosť riešenia pomocou debugovania hier s jedným hráčom, kedy bolo debugovanie značne jednoduchšie.

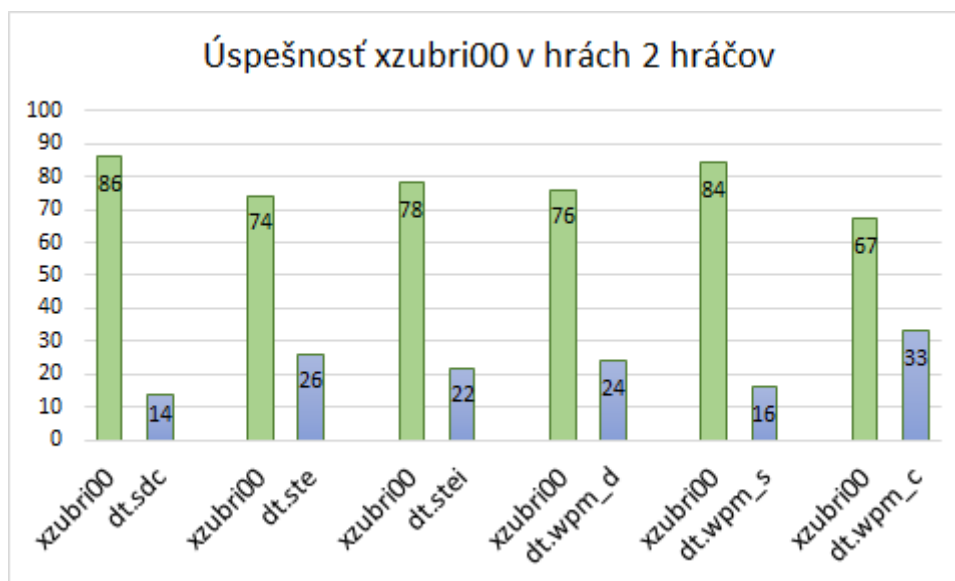
Následne sme pokračovali zisťovaním úspešnosti agenta po vykonaných zmenách a to na základe počtu výhier v hrách s jedným aj viacerými hráčmi, aby sme overili jeho schopnosti vo viacerých typoch hier.

Testovanie odhalilo náročnosť výpočtov v rámci metódy **expectimax** a viedlo nás k zavedeniu dvoch typov prerezávania, ako bolo opísané v sekcii 2.1.1.

Keď sme obmedzili veľkosť stromu vytváraného rekurzívnym zanorovaním v metóde **expectimax**, ďalej sme sa pokúšali vylepšiť výsledky agenta v hrách zmenami evaluačnej funkcie, ktorá bola pôvodne určená len ako veľkosť najväčšej súvislej oblasti. Úspešnosť týchto zmien sme opäť overovali pomocou testovania, a to na základe počtu víťazstiev v hrách s rôznymi hráčmi a ich rôznym počtom. Evaluačnú funkciu sme nakoniec upravili do podoby, o ktorej pojednáva sekcia 3.

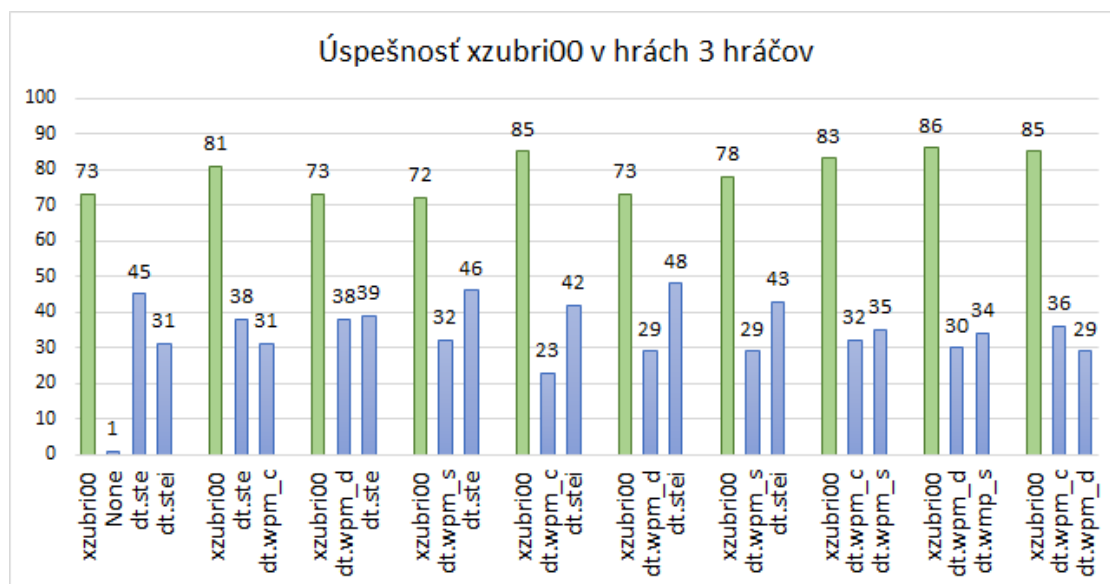
Výsledky niekoľkých záverečných testov, ktoré boli vykonané s agentom vo finálnej podobe, sú uvedené nižšie.

V grafe na obrázku 7 sú znázornené počty víťazných hier agenta **xzubri00** spoločne s počtami víťazných hier jeho rôznych spoluhráčov v hrách dvoch hráčov. V každom behu bolo spustených 100 hier. Agentovi sa podarilo s každým z vybraných protihráčov vyhrať vo viac než polovici hier. V prípade hráčov **dt.sdc**, **dt.stei**, **dt.wpm_d**, **dt.wpm_s** sa jednalo o výhry vo viac než troch štvrtinách hier.

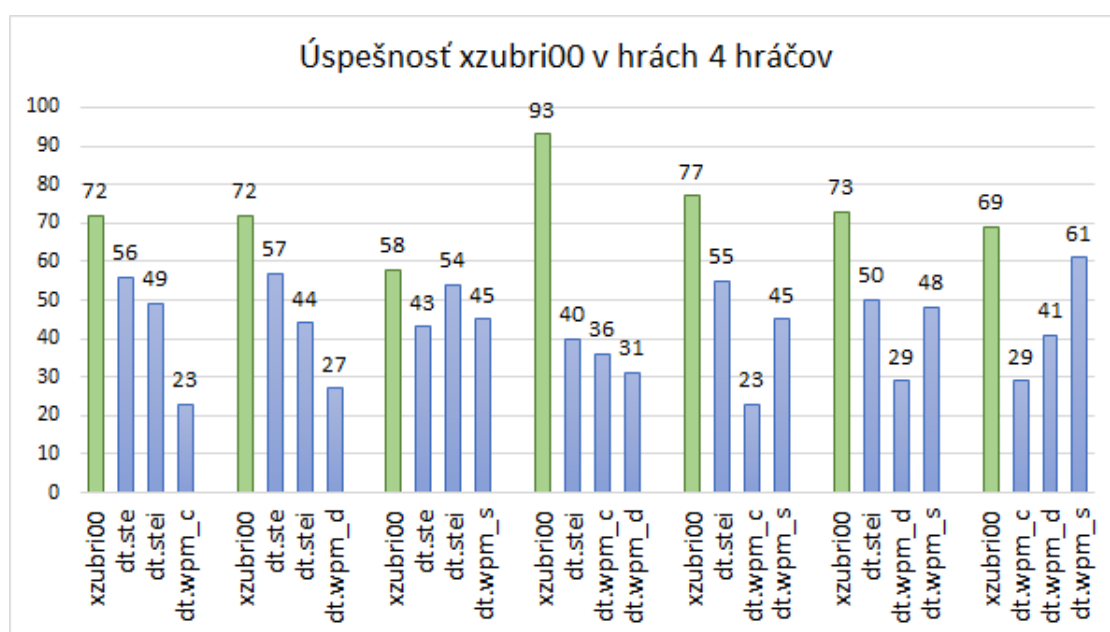


Obr. 7: Počty výhier hráča **xzubri00** v hrách 2 hráčov s rôznymi spoluhráčmi. V každom behu bolo spustených 100 hier.

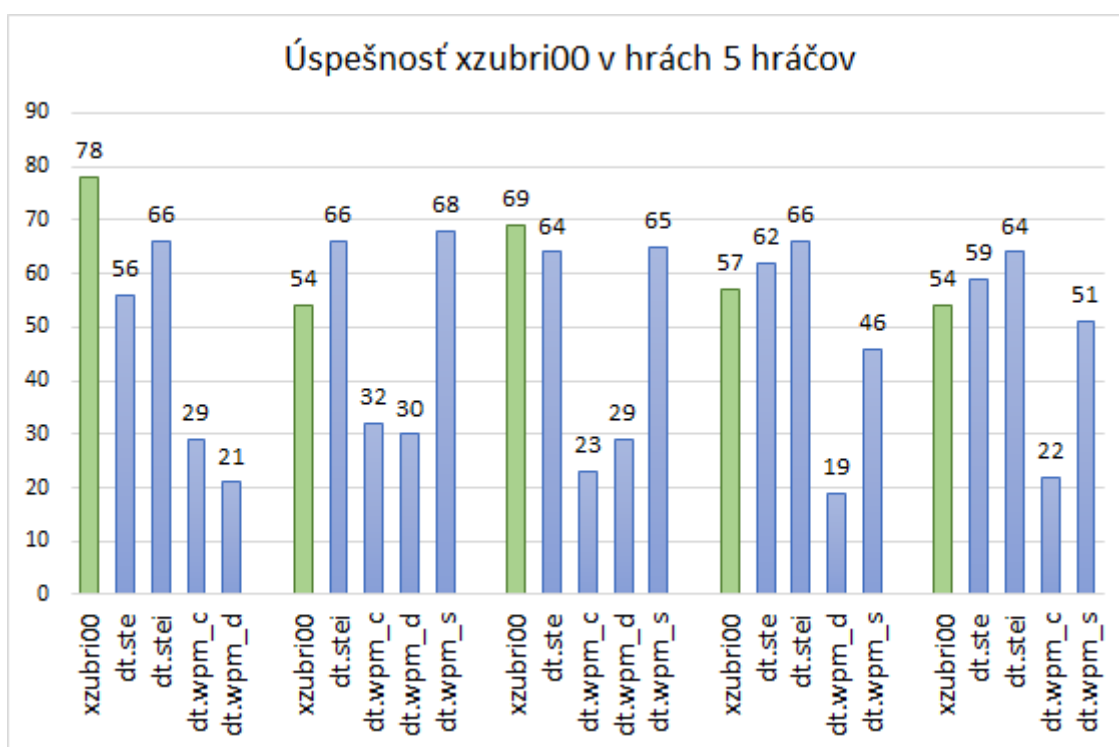
Analogicky sú na obrázkoch 8, 9, 10 zobrazené grafy s počtami víťazných hier agenta **xzubri00** s rôznymi kombináciami súperov v hrách troch, štyroch a piatich hráčov. Agent bol v hrách troch a štyroch hráčov vo všetkých z vybraných kombinácií protihráčov úspešnejší, ako ktorýkoľvek z nich. V hrách piatich hráčov však podobnú úspešnosť nedosiahol a v určitých kombináciách protihráčov sa umiestnil na tretej pozícii vzhľadom na počet víťazných hier.



Obr. 8: Počty výhier hráča xzubri00 v hrách 3 hráčov s rôznymi spoluhráčmi. V každom behu bolo spustených 150 hier.



Obr. 9: Počty výhier hráča xzubri00 v hrách 4 hráčov s rôznymi spoluhráčmi. V každom behu bolo spustených 200 hier.



Obr. 10: Počty výhier hráča xzubri00 v hrách 5 hráčov s rôznymi spoluhráčmi. V každom behu bolo spustených 250 hier.