

Pathology in Minimax Searching

Boštjan Kaluža, Mitja Luštrek, Matjaž Gams, Aleš Tavčar

Odsek za inteligentne sisteme, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana

E-pošta: bostjan.kaluza@ijs.si, mitja.lustrek@ijs.si, matjaz.gams@ijs.si, atavcar@gmail.com

Game-playing programs based on minimax principle usually produce better results with deeper searching. Theoretical analysis indicates that under specific conditions minimax behaves pathologically – deeper search produce worse results.

This paper is focused on three principal parameters of minimax searching: branching, granularity and similarity. So far existing research is focused on one of the parameters, so the main contribution of this paper is simultaneity examination of complete space of parameters. We show that pathology is rising with increasing branching factor and falling with increasing granularity and similarity degree. Different similarity models are made showing presence of pathology and examining area of partly similar trees.

1 Uvod

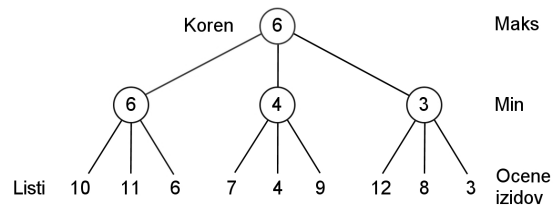
Minimaks je algoritem, na katerem temelji večina programov za igranje iger. Poskuša posnemati razmišljanje nasprotnojučih si igralcev, ki želijo v vsakem koraku odigrati najugodnejšo potezo. Pri igranju igre se zgradi drevo igre, kjer vozlišča predstavljajo možne položaje v igri, povezave so poteze, v listih pa so možni zidovi igre. Izgradnja celotnega drevesa je običajno prezamudna, zato se drevo zgradi le do globine, ki jo dopušča čas, nakar se v listih izračunajo hevristične ocene izidov.

V korenu drevesa je na potezi prvi igralec, poimenujmo ga Maks, ki želi izbrati najboljšo potezo, zato korenu priredimo maksimum vrednosti sinov. Za ovrednotenje sinov pa je potrebno vrednotiti vozlišča na nivoju 1. Sedaj je na potezi nasprotni igralec, poimenujmo ga Min, ki izbira za igralca Maks slabe poteze, zato vozliščem priredimo minimum sinov. Nivoji min in maks se izmenjujejo do listov, kjer so prave vrednosti položajev. Vrednosti v listih sta lahko le dve, npr. 1 - zmagala in 0 - poraz, lahko jih je pa tudi več. Vrednosti se nato preko min in maks nivojev prenesejo v koren drevesa. Na sliki 1 so prikazana maks in min vozlišča, izbrana poteza v korenu drevesa je veja, ki vodi k sinu z vrednostjo 6.

Praksa kaže, da globlje ko drevo igre z minimaksom preiščemo, boljše igramo, saj preiskovati globlje pomeni "premisлити" več potez naprej. Predvsem iz teoretičnih analiz pa je znano, da včasih vendarle velja ravno nasprotno: ocene izidov igre, ki so z minimaksom proti korenu prenesene z globljim preiskovanjem, so manj zanesljive od tistih, ki so rezultat plitvejšega.

Za potrebe raziskave smo globino drevesa označili z d - depth, faktor vejitve drevesa z b - branching, število različnih vrednosti (zrnatost) z g - granularity in stopnjo odvisnosti s s - similarity. Drevesa z odvisnostjo $s=0$

imenujemo neodvisna drevesa in so zgrajena povsem naključno. Drevesa z odvisnostjo $s=1$ imenujemo



Slika 1. Minimaks drevo igre

odvisna drevesa, njihovo zgradbo pa si bomo ogledali v nadaljevanju. Odvisno drevo je mogoče zgraditi na več načinov, njihova skupna lastnost je pa ta, da imajo bratski listi podobne vrednosti. V nadaljevanju bomo predstavili nekaj načinov gradnje dreves, saj se je izkazalo, da na obnašanje modela vpliva način gradnje odvisnih dreves. Delno odvisno drevo se zgradi kot kombinacijo odvisnega in neodvisnega drevesa glede na parameter s . Pri izgradnji zgradimo tako neodvisno kot odvisno drevo in ju nato združimo v razmerju $neodvisno : odvisno = (1-s) : s$.

Na podlagi raziskav [1, 2] smo postavili hipotezo, da se patologija povečuje z naraščanjem vejitve drevesa igre in pada s povečevanjem števila vrednosti v vozliščih ter njihovo odvisnostjo. Naš cilj je preiskati celoten prostor parametrov v okviru časovnih zmožnosti (v smislu procesiranja) ter raziskati in pokazati njihovo odvisnost.

2 Neodvisni model

Model najprej zgradi drevo, z minimaksom izračuna prave vrednosti in si jih zapomni. Napako hevristične funkcije simulira tako, da prave vrednosti na izbranem nivoju zašumi z Gaussovimi šumom s standardnim odklonom $\sigma=0,1$ in jih z minimaksom prenese v koren drevesa. Zašumljene vrednosti smo poimenovali hevristične vrednosti.

Kot mero napake smo vzeli delež napačnih potez pri izbrani globini preiskovanja. Pri ocenjevanju napake se najprej zabeležijo veje s pravimi vrednostmi, pri katerih se pojavijo maksimalne vrednosti sinov. Take veje poimenujemo prave poteze. Nato se drevo zašumi na izbranem nivoju in pomožne vrednosti se prenesejo v koren. Tokrat opazujemo veje z maksimalnimi hevrističnimi vrednostmi in izračunamo delež vej, ki se ujemajo s pravimi potezami. Dobljena vrednost predstavlja delež pravih potez pri izbrani globini preiskovanja, zato jo je potrebno odšteti od 1. Pri

izbrani globini i smo delež napačnih potez označili z $error_i$.

Do patologije pride, ko je delež napačnih potez pri preiskovanju do nižjega nivoja večji od deleža napačnih potez pri preiskovanju do višjega nivoja. V meritvah smo opazovali napaki pri preiskovanju do prvega in petega nivoja, patologijo smo označili s p - pathology.

$$p = \frac{error_5}{error_1} \quad (1)$$

Patologije je prisotna, ko je $p > 1$, kar pomeni, da smo pri globljem preiskovanju naredili več napačnih potez.

Model je bil v prvem koraku zgrajen po zgledu Luštrekovega modela [2] in omogoča preiskovanje prostora zrnatosti in vejitve. Neodvisno drevo se zgradi z generiranjem naključnih vrednosti v listih, pri tem pa je potrebno zagotoviti, da položaji v drevesu niso preveč v prid enemu od igralcev, saj bi bila igra že odločena. Verjetnost zmage in poraza smo uravnavali s parametrom c_b , ki so ga opisali že Pearl, Nau, Bratko in Gams.

Zrnatost smo generirali tako, da smo interval $[0,1]$ razdelili na g podintervalov enake dolžine. Posamezna vrednost je predstavljena s srednjo vrednostjo podintervala.

Rezultati meritev patološkosti so skladni z obstoječimi rezultati [2] in so povzeti v tabeli 1. V njej so zapisane mejne vrednosti zrnatosti, pri kateri patologija izgine, se poslabša za 2% in izboljša za 2%. Vsaka meritev je pridobljena s simulacijo Monte Carlo z 10000 drevesi.

$p \setminus b$	2	3	4	5	6	7	8	9
0,98	10	12	16	20	26	31	40	46
1	11	13	18	23	30	39	58	85
1,02	12	14	20	26	41	61	90	152

Tabela 1. Meja patologije za različne vejitve neodvisnega drevesa

3 Odvisna drevesa

Pri vpeljavi odvisnosti smo za njeno merjenje uporabili faktor grozdenja cf - clustering factor povzet po Sadikovu [3] in je količnik med povprečnim standardnim odklonom sinov in standardnim odklonom celotnega drevesa. Z izračunom je mogoče zaznati, kdaj so si bližnja vozlišča nadpovprečno podobna. Pri dovolj velikem drevesu je standardni odklon celotnega drevesa neodvisen od stopnje odvisnosti vozlišč, medtem ko je povprečen standardni odklon sinov pri večji odvisnosti manjši.

Izračuna se po formuli (2), kjer je N število vseh notranjih vozlišč, M je število vseh vozlišč, v_{ij} je j -ti sin i -tega vozlišča, v_i je i -to vozlišče, av_i je povprečje sinov vozlišča i in av je povprečna vrednost vseh vozlišč.

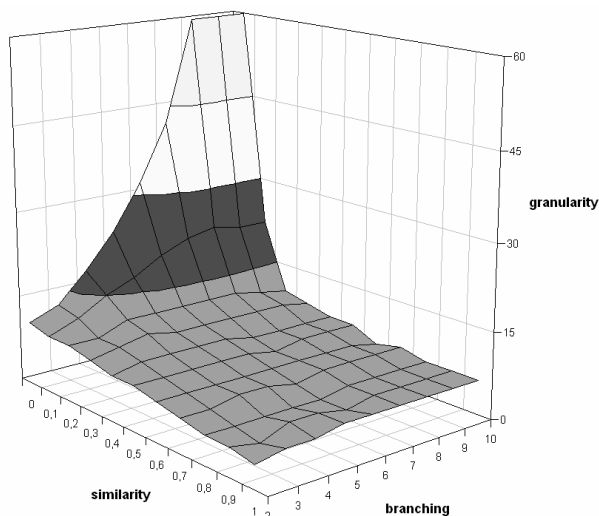
$$cf = 1 - \frac{\frac{1}{N} \sum_{i=1}^N \sqrt{\frac{1}{b} \sum_{j=1}^b (v_{ij} - av_i)^2}}{\sqrt{\frac{1}{M} \sum_{i=1}^M (v_i - av)^2}} \quad (2)$$

3.1 Porazdelitev pomožnih vrednosti

Gradnja popolnoma odvisnega drevesa poteka tako, da se korenu drevesa pripiše pomožno začetno vrednost 0,5, ki se jo nato prenaša na sinove. Pri vsakem prenosu na nižji nivo se vrednost porazdeli okrog starša z normalno porazdelitvijo. Vrednosti, ki po porazdelitvi segajo izven intervala, ponovno porazdelimo, dokler ne pristanejo v intervalu. Bratski listi se tako med seboj razlikujejo le za en nivo porazdeljevanja.

Delno urejeno drevo se zgradi tako, da se na predzadnjem nivoju z verjetnostjo s odloči ali bodo sinovi (listi) vsebovali naključne ali odvisne vrednosti.

Dobljenemu drevesu je potrebno še preslikati vrednosti listov v predstavnike zrnatosti. Pri preslikavi je potrebno upoštevati delež zmag in porazov c_b , za katerega pa se izkaže, da je pri odvisnih drevesih drugačen kot pri neodvisnih.



Slika 2. Rezultati meritev pri generiranju odvisnih dreves s porazdelitvijo pomožnih vrednosti

Izračun c_b -ja smo povzeli po izvornih izračunih, ki težijo k temu, da se verjetnosti zmage in poraza na sodih in lihih nivojih izmenjujeta, kar pomeni, da sta verjetnost na sodih in njena obratna vrednost na lihih nivojih čim bolj enaki.

Rezultati so prikazani na sliki 2, kjer je prikazana mejna ploskev patologije $p=1$ v prostoru preiskovanih parametrov: območje pod ploskvo je patološko, nad ploskvo pa nepatološko.

Nekoliko nas je presenetilo dejstvo, da patologija povsem ne izgine pri popolnoma odvisnih drevesih. Pričakovali smo, da bo pri povečevanju odvisnosti patologija upadala, kar se tudi zgodi, in pri popolni

odvisnosti izginila. Odločili smo se, da bomo preizkusili še nekaj načinov gradnje odvisnih dreves.

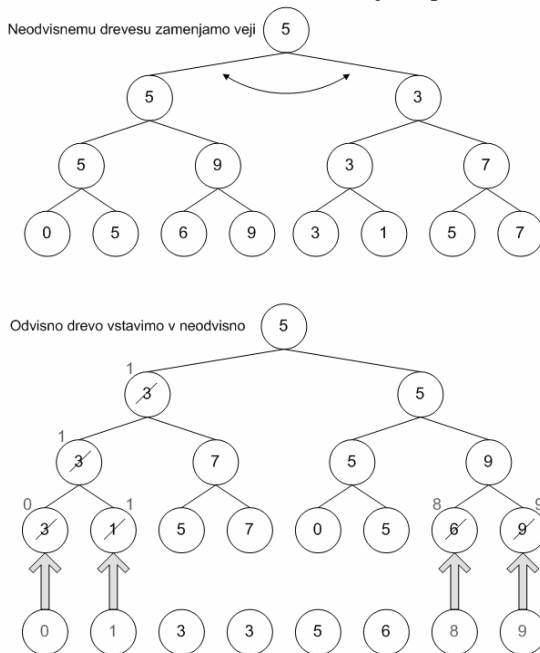
3.2 Urejanje naključnih vrednosti

Odvisnim drevesom se v listih dodelijo naključne vrednosti, nato pa se sortirajo po velikosti. Listi imajo tako po naraščajočih indeksih naraščajoče vrednosti. Pri preslikavi vrednosti v predstavnike zrnatosti smo za izračun razmerja c_b uporabili postopek, kjer se verjetnost zmage in poraza na sodih in lihah nivojih izmenjujeta.

Delno odvisno drevo smo zgradili z naključnim vstavljanjem listov odvisnega drevesa v neodvisno glede na verjetnost s . Težavo je povzročala stopnja odvisnosti $s=0,6$, kjer je bila patologija prisotna pri vseh zrnatostih. Domnevali smo, da razlog za takšno obnašanje tiči v ponesrečeni gradnji delno odvisnih dreves. Če liste odvisnega drevesa dodajamo listom neodvisnega po naraščajočem vrstnem redu, pri tem z večanjem odvisnosti povečamo verjetnost prave odločitve zadnje (desne) veje. V primeru, ko zadnja veja ne vsebuje prave odločitve, favoriziramo napačno vejo in s tem povečamo verjetnost patologije. Pojav bi lahko omilili tako, da bi veje neodvisnega drevesa obravnavali v naraščajočem vrstnem redu.

3.2.1 Delno urejanje neodvisnega drevesa

V naslednjem koraku smo pred gradnjo delno odvisnih dreves uredili veje neodvisnega drevesa na prvem nivoju. Naključna porazdelitev vrednosti v listih je ostala nespremenjena, saj smo le veje obravnavali v drugačnem vrstnem redu. Na sliki 3 je zgoraj prikazano neodvisno drevo, ki mu obrnemo veji na prvem nivoju.



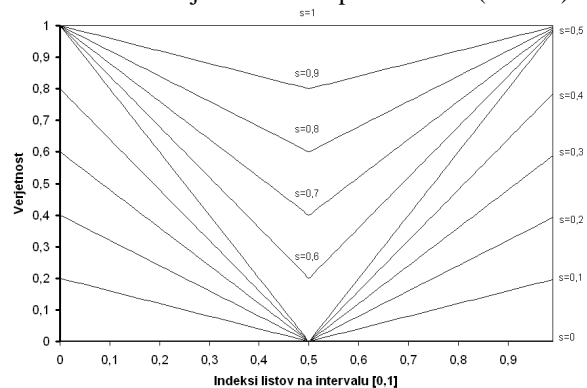
Slika 3. Generiranje delno odvisnega drevesa: neodvisnemu drevesu zamenjamo veji in del listov nadomestimo z naraščajoče urejenimi vrednostmi odvisnega drevesa

Tako urejenim vejam smo z leve in desne z verjetnostjo 1 zamenjali $s/2$ vrednosti. Na sliki 3 je spodaj prikazano neodvisno drevo z zamenjanimi vejami, pod njim pa so listi odvisnega drevesa z naraščajoče urejenimi vrednostmi, ki jih z leve in desne vstavimo na pripadajoča mesta v neodvisno drevo. Dodajanje na ta način zagotavlja, da se izognemo zgoraj opisanemu problemu favoriziranja napačne veje, saj je v tem primeru prava (maksimalna) veja vedno na levi strani. Poveča se le odvisnost v listih.

Stalna prisotnost patologije pri stopnji odvisnosti $s=0$ je izginila, stranski učinek pa je bila visoka odvisnost delno odvisnih dreves. Patologija je namreč povsem izginila že pri odvisnosti $s=0,4$. Takšna odvisnost je nekoliko "premočna", da bi lahko opazovali, kaj se dogaja z delno odvisnimi drevesi, zato smo sklenili gradnjo delno odvisnih dreves prilagoditi tako, da bi prehod iz popolnoma odvisnih v neodvisna drevesa omilili.

3.2.2 Trikotniška porazdelitev

Gradnjo delno odvisnih dreves z dodajanjem z leve in desne strani smo poimenovali kvadratna porazdelitev. Za ublažitev prehoda bi potrebovali porazdelitev, ki prehaja bolj postopoma med dvema skrajnima drevesoma. Taka je trikotniška porazdelitev (Slika 4).



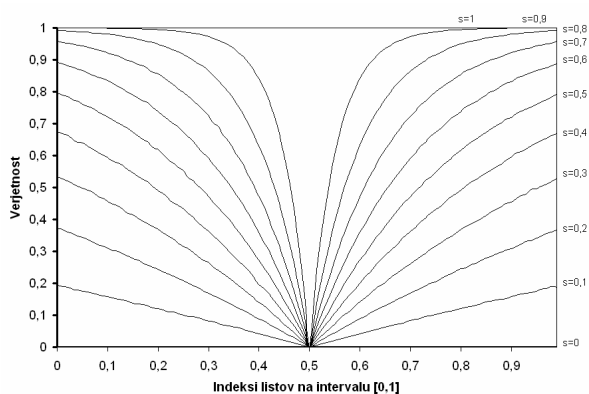
Slika 4. Trikotniška porazdelitev $H_s(i)$

Definirali smo funkcijo $H_s(i)$, ki predstavlja verjetnost, da bo v neodvisnem drevesu vrednost v i -tem listu zamenjana z vrednostjo pripadajočega indeksa odvisnega drevesa pri stopnji odvisnosti s .

Povprečna višina obeh delov trikotnika je enaka stopnji odvisnosti s . Pri stopnji odvisnosti $s>0,5$ se srednja točka med trikotnikoma dvigne, tako da je povprečna višina še vedno s . Višina pri posameznem indeksu i predstavlja verjetnost $H_s(i)$. Pri simulaciji velikega števila zgrajenih dreves verjetnost zamenjave posameznega indeksa konvergira k verjetnosti $H_s(i)$. Pri stopnji odvisnosti $s=1$ so zamenjani vsi elementi neodvisnega drevesa in pri stopnji $s=0$ neodvisno drevo ostane nespremenjeno. Meritve so pokazale zmernejši prehod med odvisnimi in neodvisnimi drevesi, vendar le do odvisnosti $s=0,5$. Nezveznost same trikotniške porazdelitve pri $s=0,5$ se namreč odraža tudi v rezultatih.

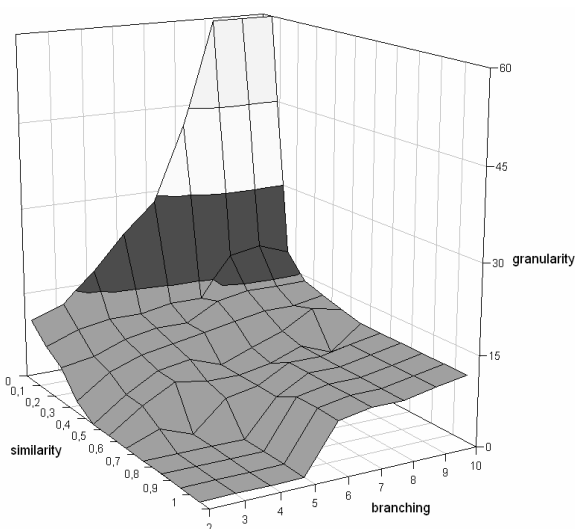
3.2.3 Gladka porazdelitev

Porazdelitveno funkcijo smo spremenili tako, da smo uporabili $H_{0,5}(i)$ iz trikotniške porazdelitve in dodajali liste odvisnega drevesa v neodvisno dokler drevo ni vsebovalo s odvisnih vozlišč. Tako smo dosegli, da se je več vrednosti dodalo na robove. Zaradi hitrejše simulacije, smo porazdelitev pri posamezni stopnji odvisnosti izračunali v ločeni simulaciji in med gradnjo uporabili izračunane vrednosti. Gladka porazdelitev $H_s(i)$ je za različne stopnje s prikazana na sliki 5.



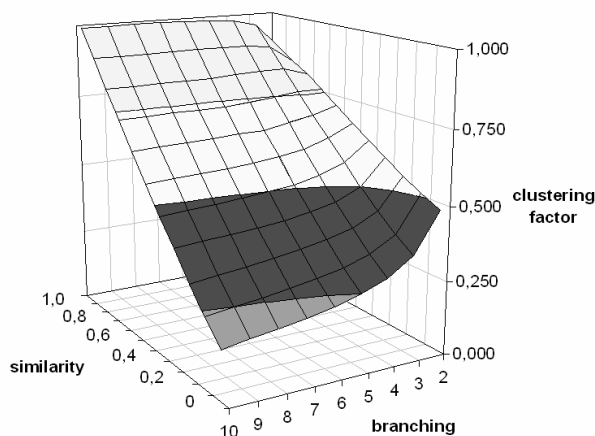
Slika 5. Gladka porazdelitev $H_s(i)$

Rezultat je zvezen prehod med neodvisnimi in odvisnimi drevesi, rezultati meje patologije pri $p=1$ so prikazani na sliki 6.



Slika 6. Rezultati meritev pri generiranju odvisnih dreves s sortiranjem naključnih vrednosti in vstavljanjem listov v neodvisno drevo z gladko porazdelitvijo

Patologija pri popolni odvisnosti in pri manjših vejitvah popolnoma izgine, medtem ko je pri večjih vejitvah kljub odvisnosti venomer prisotna. Po formuli (2) smo izmerili faktor grozdenja za celoten prostor generiranih dreves, ki je prikazan na sliki 7.



Slika 7. Faktor grozdenja generiranih dreves pri gladki porazdelitvi

Faktor grozdenja pri posamezni vejitvi linearno pada z zmanjševanjem stopnje odvisnosti in je nekoliko višji pri manjših vejitvah. Pri večji odvisnosti so bratska vozlišča urejena naraščajoče, kar pomeni da so si njihove vrednosti podobne in posledično je povprečni standardni odklon med brati manjši. Višji faktor grozdenja pri manjših vejitvah pa je posledica manjšega števila vozlišč, s katerimi računamo standardni odklon.

4 Zaključek

Potrdili smo hipotezo, da stopnja patologije upada s povečevanjem odvisnosti ter znatosti in narašča s povečevanje vejitve drevesa. Preiskali smo celoten prostor parametrov in pokazali, da je pojav prisoten tudi pri različnih načinih gradnje odvisnih dreves (slika 2, slika 6).

V nadaljnjem delu bomo preizkusili obnašanje realnih iger, kot sta šah ter Pearlova in Nauova igra, in jih umestili v model. Med preizkusi smo opravljali tudi meritve na drevesih minimin, ki kažejo, da je pojav patologije prisoten tudi pri tem tipu dreves. Opravili bomo nadaljnje meritve in tudi te preizkusili na realnih problemih.

5 Literatura

- [1] Luštrek, M., Gams, M., and Bratko, I. (2006). Is Real-Valued Minimax Pathological? Artificial Intelligence 170 (6-7).
- [2] Mitja Luštrek, Patologija v hevrstičnih preiskovalnih algoritmih. Doktorska disertacija, Univerza v Ljubljani, 2007.
- [3] Aleksander Sadikov, Širjenje napak hevrstičnih ocen v preiskovanju grafov iger. Doktorska disertacija, Univerza v Ljubljani, 2005.
- [4] Luštrek, M., Bratko, I., and Gams, M. (2005). Why Minimax Works: An Alternative Explanation. IJCAI conference.