

Selective search in games of different complexity

Citation for published version (APA):

Schadd, M. P. D. (2011). *Selective search in games of different complexity*. [Doctoral Thesis, Maastricht University]. Maastricht University. <https://doi.org/10.26481/dis.20110525ms>

Document status and date:

Published: 01/01/2011

DOI:

[10.26481/dis.20110525ms](https://doi.org/10.26481/dis.20110525ms)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Summary

This thesis investigates how selective-search methods can improve the performance of a game program for a given domain. Selective-search methods aim to explore only the profitable parts of the state space, but take the risk to overlook the best move. We propose several selective-search methods and test them in a large number of game domains.

In Chapter 1 we provide a brief introduction on games research and discuss selective search for games. The following problem statement guides our research.

Problem statement: *How can we improve selective-search methods in such a way that programs increase their performance in domains of different complexity?*

Rather than testing selective-search methods on one class of games, we chose different classes of games, which all have to be addressed differently. Each class of games represents a level of complexity. Between every level there exists a *complexity jump*. With a complexity jump the complexity of the game increases significantly because the mechanism of the game is changed. We have chosen five different levels of games, resulting in four complexity jumps. (1) One-player games, or puzzles, involve no opponent and are a testbed for planning algorithms. (2) Two-player games are the classic testbed for search methods. We use them for investigating mate-solvers. For testing search with chance nodes, (3) non-deterministic and (4) imperfect-information games may be used. (5) Multi-player games are a testbed for dealing with coalition forming. We formulate four research questions to guide our research. Each one deals with search for a different class of games and a different selective-search method. The four research questions address (1) Monte-Carlo Tree Search (MCTS), (2) Proof-Number (PN) search, (3) expectimax, and (4) multi-player search.

Chapter 2 is a general introduction to search methods for games. It explains the minimax algorithm and the well-known $\alpha\beta$ search. Standard techniques for enhancing the $\alpha\beta$ search are discussed as well. We furthermore explain MCTS and its enhancements.

The traditional approach to deterministic one-player games with perfect information is applying A* or IDA*. These methods have been quite successful in coping

with this class of games. The disadvantage of these methods is that they require an admissible heuristic evaluation function. The construction of such a function can be difficult. Since the selective-search method MCTS does not require an admissible heuristic, it may be an interesting alternative. This has led us to the first research question.

Research question 1: *How can we adapt Monte-Carlo Tree Search for a one-player game?*

Chapter 3 answers the first research question by proposing a new MCTS variant called Single-Player Monte-Carlo Tree Search (SP-MCTS). MCTS is adapted by two modifications resulting in SP-MCTS. The modifications concern (1) the selection strategy and (2) the backpropagation strategy. The selection strategy is adapted to take the standard deviation of the scores for a move into account and to combine the average score with the top score. The backpropagation strategy is modified to provide the selection strategy with all required information. For evaluating SP-MCTS, the puzzle SameGame is chosen as test domain. So far, there does not exist a good admissible heuristic evaluation function for this game.

On the standardized test set of 20 SameGame positions, the manually tuned SP-MCTS method, which invests all search time at the initial position, scored 73,998 points. This was the highest score on the test set at that point of time (2008). The main contribution is therefore that we successfully adapted MCTS for a one-player game. Inspired by our approach, two other Monte-Carlo-based approaches, Nested Monte-Carlo and Heuristically Guided Swarm Tree Search, broke our record subsequently. At the time of publishing this thesis SP-MCTS, with parameters tuned by the Cross-Entropy Method and with time equally distributed over the consecutive positions, scored 78,012 points on the test set, which is currently the third highest score (2010). Thus, answering research question 1, we have shown that MCTS is applicable to a one-player deterministic perfect-information game. Our variant, SP-MCTS, is able to achieve good results in the game of SameGame. SP-MCTS is a worthy alternative for puzzles where a good admissible estimator cannot be found.

Ideally, a search method is able to prove that a move is the optimal one for a given game. The game is solved if this is achieved. A selective-search method specially designed for solving (end)games is Proof-Number (PN) search. PN search is efficient in searching game trees with a non-uniform branching factor. Moreover, for quite some games, endgame databases played a role in solving. This has led us to the second research question.

Research question 2: *How can we solve a two-player game by using Proof-Number search in combination with endgame databases?*

Chapter 4 answers the second research question by examining the tradeoff between time spent on PN search and time spent on creating endgame databases when solving the game of Fanorona. This game has a material-based theme with a state-space complexity similar to checkers. Endgame-database statistics show that (1) the player to move has an advantage and (2) that a draw can often be achieved in

spite of having fewer pieces than the opponent. The optimal endgame-database sizes for the 3×9 , 5×5 , and 7×5 Fanorona variants are 3, 4, and 5 pieces, respectively. We conclude that the optimal database size is located at the point where the time required for database construction and the time required for solving by PN search are of the same order. Our main result is that standard Fanorona (5×9) has been weakly solved and is drawn when both players play optimally, adding Fanorona to the list of solved games. This result was achieved by combining the PN-search variant PN^2 with all endgame databases up to 7 pieces.

Another form of searching selectively in two-player deterministic games with perfect information is variable-depth search. Branches can be pruned if they seem unpromising (forward pruning), or extended if the branches are promising (search extensions). There exist several successful forward-pruning techniques for the $\alpha\beta$ algorithm. For two-player games with non-determinism or imperfect information expectimax may be used. Expectimax adds chance nodes to the search tree. There are, however, no forward-pruning techniques available for chance nodes. This has led us to the third research question.

Research question 3: *How can we perform forward pruning at chance nodes in the expectimax framework?*

Chapter 5 answers the third research question by proposing the forward-pruning technique ChanceProbCut for expectimax. This technique is the first in its kind to forward prune at chance nodes. ChanceProbCut is inspired by the $\alpha\beta$ forward-pruning technique ProbCut. ChanceProbCut estimates values of chance events based on shallow searches. Based on the correlation between evaluations obtained from searches at different depths, ChanceProbCut prunes chance events in advance if the result of the chance node probably falls outside the search window. Two non-deterministic games (Dice and ChanceBreakthrough) and a game of imperfect information (Stratego) served as test domains. Experiments revealed that ChanceProbCut is able to reduce the size of the game tree significantly without a loss of decision quality in Stratego, Dice, and ChanceBreakthrough. A safe node reduction of between 30% and 85% is achieved across all games. Thus, ChanceProbCut finds the best move faster in the expectimax framework, while not affecting the playing strength. The gained time may be invested in a deeper search. Selfplay experiments in Stratego and Dice showed that there is a small but relevant improvement in playing strength. In ChanceBreakthrough, though, a significant increase in performance was measured. ChanceProbCut is able to win 54.4% on 4,000 games.

In deterministic two-player games with perfect information, the majority of research focused on the $\alpha\beta$ algorithm. For deterministic multi-player games with perfect information, the choice of algorithm is not as straightforward. The two main algorithms are called \max^n and paranoid, both approaching the problem from a different angle. \max^n assumes that every player tries to maximize the own score, while paranoid assumes that all opponents form a coalition against the root player. However, these assumptions have drawbacks. Due to the lack of safe pruning in \max^n only a limited lookahead is possible. Furthermore, the underlying assumption

of \max^n may be unrealistic, resulting in \max^n to be too optimistic. When searching deep with the paranoid algorithm, the other players may dominate the root player, resulting in paranoid to be too pessimistic. This has led us to the fourth research question.

Research question 4: *How can we improve search for multi-player games?*

Chapter 6 answers the fourth research question by proposing a new search algorithm, called Best-Reply Search (BRS), for deterministic non-cooperative multi-player games with perfect information. This algorithm allows only one opponent to play a counter move. This opponent is the one with the strongest move against the root player. The other players have to pass their turn. Using this approach, more turns of the root player can be searched, resulting in long-term planning. At the same time, some sort of cautiousness is preserved by searching the strongest opponent move.

We have chosen three deterministic multi-player games of perfect information, i.e., Chinese Checkers, Focus, and Rolit. BRS is able to significantly outperform \max^n in these games, with a win ratio of between 65% and 95%. Against paranoid, BRS is significantly stronger in Chinese Checkers and Focus, with win ratios of between 57% and 71%. In Rolit, BRS and paranoid are on equal footing. When playing different kind of opponents at the same time, BRS is the strongest algorithm in Chinese Checkers and Focus. In Rolit, BRS is somewhat behind paranoid. Increasing the search time generally does not have a negative effect on the performance of BRS. This implies that searching illegal positions, which are generated by forcing opponents to pass, does not have a large influence. The possible negative effect is outbalanced by the larger lookahead.

Chapter 7 concludes the thesis and gives an outlook on open questions and directions for future research. Taking the answers to the research questions above into account we see that there are four ways to improve selective-search methods. First, Single-Player Monte-Carlo Tree Search balances exploitation and exploration such that it is a worthy alternative for one-player games where a good admissible estimator cannot be found. Second, PN search with endgame databases, which prefers narrow subtrees above wide ones, is able to prove that the game-theoretic value of the two-player game Fanorona is a draw. Third, ChanceProbCut can forward prune chance events in two-player games with non-determinism or imperfect information. Fourth, in non-cooperative deterministic multi-player games with perfect information, Best-Reply Search achieves long-term planning by assuming that only one opponent is allowed to play a counter move.

While this thesis shows that selective-search methods are successful in quite some domains, all our proposed enhancements and algorithms can be tested in other domains, as well. These domains include classes of games with corresponding complexity levels, but also game classes which have not been covered in this research. We mention one-player games with non-determinism, two-player games with non-determinism and imperfect information, and multi-player games with non-determinism and/or imperfect information.

Samenvatting

Dit proefschrift onderzoekt hoe selectieve zoekmethoden de prestaties van een spelprogramma kunnen verbeteren voor een bepaald domein. Selectieve zoekmethoden hebben als doel om alleen de winstgevendste delen van de zoekruimte te verkennen, maar ze nemen het risico om de beste zet te overzien. Wij stellen een aantal selectieve zoekmethoden voor en testen ze in een groot aantal speldomeinen.

In hoofdstuk 1 geven we een korte inleiding in onderzoek in spelen en bespreken selectieve zoekmethoden voor spelen. De volgende probleemstelling stuurt ons onderzoek.

Probleemstelling: *Hoe kunnen we selectieve zoekmethoden op een zodanige wijze verbeteren dat programma's hun prestaties in domeinen van verschillende complexiteit verhogen?*

In plaats van selectieve zoekmethoden te testen op één klasse van spelen, kozen we verschillende klassen van spelen, die allemaal anders aangepakt moeten worden. Elke spelklasse vertegenwoordigt een niveau van complexiteit. Tussen elk niveau bestaat er een *complexiteitsprong*. Met een complexiteitsprong wordt de complexiteit van het spel aanzienlijk verhoogd, omdat het mechanisme van het spel is veranderd. We hebben gekozen voor vijf verschillende niveaus van spelen, wat resulteert in vier complexiteitsprongen. (1) Éénspeler spelen, of puzzels, hebben geen tegenstander en zijn een testdomein voor planningalgoritmes. (2) Tweespeler spelen zijn het klassieke testdomein voor zoekmethoden. We gebruiken ze voor zoekmethoden die de speltheoretische waarde kunnen bewijzen. Voor zoeken met kansknopen kunnen (3) niet-deterministische en (4) imperfecte-informatie spelen worden gebruikt. (5) Meerspeler spelen zijn een testdomein voor het omgaan met coalitievorming. We formuleren vier onderzoeksvragen die ons onderzoek sturen. Elke vraag heeft als thema het zoeken in een andere klasse van spelen en een andere selectieve zoekmethode. De vier onderzoeksvragen gaan over (1) Monte-Carlo Tree Search (MCTS), (2) Proof-Number (PN) search, (3) expectimax, en (4) meerspeler zoekmethoden.

Hoofdstuk 2 is een algemene inleiding in zoekmethoden voor spelen. Het beschrijft het minimax algoritme en de bekende $\alpha\beta$ zoekmethode. Standaardtechnieken voor het verbeteren van de $\alpha\beta$ zoekmethode worden ook besproken. Verder leggen wij MCTS en de bijbehorende verbeteringen uit.

De traditionele benadering voor deterministische éénpeler spelen met perfecte informatie is het toepassen van A* of IDA*. Deze methoden zijn zeer succesvol in het omgaan met deze klasse van spelen. Het nadeel van deze methoden is dat ze een onderschattende evaluatiefunctie vereisen. Het maken van een dergelijke functie kan moeilijk zijn. Omdat de selectieve zoekmethode MCTS geen evaluatiefunctie nodig heeft, is het een mogelijk interessant alternatief. Dit heeft geleid tot de eerste onderzoeksvraag.

Onderzoeksvraag 1: *Hoe kunnen we Monte-Carlo Tree Search aanpassen voor een éénpeler spel?*

Hoofdstuk 3 geeft antwoord op de eerste onderzoeksvraag door een nieuwe MCTS variant voor te stellen, genaamd Single-Player Monte-Carlo Tree Search (SP-MCTS). Twee wijzigingen in MCTS resulteren in SP-MCTS. De wijzigingen hebben betrekking op (1) de selectiestrategie en (2) de propagatiestrategie. De selectiestrategie is aangepast door rekening te houden met de standaarddeviatie van de scores van een zet en door de gemiddelde score te combineren met de hoogste score. De propagatiestrategie is aangepast om de selectiestrategie van alle benodigde informatie te voorzien. Voor de evaluatie van SP-MCTS, is de puzzel SameGame gekozen als testdomein. Tot dusver bestaat er geen goede onderschattende heuristische evaluatiefunctie voor dit spel.

Voor 20 gestandaardiseerde SameGame testposities scoorde de handmatig ingestelde SP-MCTS methode, waarbij alle zoektijd geïnvesteerd werd aan het begin van het spel, 73.998 punten. Dit was de hoogste score op de testposities in 2008. De belangrijkste bijdrage is dan ook dat wij met succes MCTS aangepast hebben voor een éénpeler spel. Geïnspireerd door onze aanpak, hebben twee andere Monte-Carlo gebaseerde methoden, Nested Monte-Carlo Search en Heuristically Guided Swarm Tree Search, ons record later verbroken. Op het moment van publicatie van dit proefschrift heeft SP-MCTS, met parameters afgesteld door de Cross-Entropy Method en met de tijd gelijkmatig verdeeld over alle zetten, 78.012 punten gescoord op de testposities, wat momenteel de derde hoogste score is (2010). Dus, om onderzoeksvraag 1 te beantwoorden, we hebben aangetoond dat MCTS toepasbaar is op een deterministisch éénpeler spel met perfecte informatie. Onze variant, SP-MCTS, is in staat om goede resultaten te behalen in SameGame. SP-MCTS is een waardig alternatief voor puzzels waar geen goede onderschattende evaluatiefunctie gevonden kan worden.

Idealiter is een zoekmethode in staat om te bewijzen dat een zet optimaal is voor een bepaald spel. Het spel is opgelost als dit wordt bereikt. Een selectieve zoekmethode speciaal ontworpen voor het oplossen van (eind)spelen is Proof-Number (PN) search. PN search is efficiënt in het doorzoeken van spelbomen met een niet-uniforme vertakkingsgraad. Daarnaast hebben voor veel spelen eindspeldatabases een rol gespeeld bij het oplossen. Dit heeft geleid tot de tweede onderzoeksvraag.

Onderzoeksvraag 2: *Hoe kunnen we een tweespeler spel oplossen met behulp van Proof-Number search in combinatie met eindspeldatabases?*

Hoofdstuk 4 geeft antwoord op de tweede onderzoeksvraag door het onderzoeken van de verhouding tussen de tijd besteed aan PN search en de tijd besteed aan het creëren van de eindspeldatabases bij het oplossen van het spel Fanorona. Dit spel is qua complexiteit van de toestandsruimte vergelijkbaar met Engels dammen. Uit statistieken van de eindspeldatabases blijkt dat (1) de speler aan zet een voordeel heeft en (2) dat ook met minder stukken op het bord dan de tegenstander een remise vaak kan worden gerealiseerd. De optimale grootten van de eindspeldatabase voor de 3×9 , 5×5 , en 7×5 Fanorona varianten zijn respectievelijk 3, 4, en 5 stukken. We concluderen dat de optimale grootte van de database op het punt ligt waar de tijd die nodig is om de database uit te rekenen en de tijd die nodig is voor het oplossen met PN search van dezelfde orde zijn. Ons belangrijkste resultaat is dat standaard Fanorona (5×9) is opgelost. Het resultaat is een remise als beide spelers optimaal spelen. Fanorona is dus toegevoegd aan de lijst van opgeloste spelen. Dit resultaat werd bereikt door het combineren van de PN-search variant PN^2 met alle eindspeldatabases tot 7 stukken.

Een andere vorm van selectieve zoekmethoden in deterministische tweespeler spelen met perfecte informatie is variable-depth search. Takken kunnen worden gesnoeid als ze weinig belovend lijken (voorwaarts snoeien), of verlengd indien de takken veelbelovend zijn (zoekextensies). Er bestaan een aantal succesvolle voorwaarts snoeitechnieken voor de $\alpha\beta$ zoekmethode. Voor tweespeler spelen die niet deterministisch zijn of imperfecte informatie hebben kan expectimax worden gebruikt. Expectimax voegt kansknopen toe aan de zoekboom. Er zijn echter geen voorwaarts snoeitechnieken beschikbaar voor de kansknopen. Dit heeft geleid tot de derde onderzoeksvraag.

Onderzoeksvraag 3: *Hoe kunnen we voorwaarts snoeien in de kansknopen van expectimax?*

Hoofdstuk 5 geeft antwoord op de derde onderzoeksvraag door de voorwaarts snoeitechniek ChanceProbCut te beschrijven voor expectimax. Deze techniek is de eerste in zijn soort om voorwaarts te snoeien in kansknopen. ChanceProbCut is geïnspireerd door de voorwaarts snoeitechniek ProbCut voor $\alpha\beta$. ChanceProbCut schat waarden van kansgebeurtenissen gebaseerd op ondiepe zoekopdrachten. Gebaseerd op een correlatie tussen evaluaties van zoekopdrachten met verschillende dieptes, kan ChanceProbCut kansknopen voorwaarts snoeien als het resultaat van de kansknoop waarschijnlijk buiten het zoekvenster valt. Twee niet-deterministische spelen (Dice en ChanceBreakthrough) en één imperfecte-informatie spel (Stratego) dienden als testdomeinen. De experimenten laten zien dat ChanceProbCut in staat is om grote delen van de zoekboom zonder kwaliteitsverlies te snoeien. Een veilige reductie in knopen van tussen de 30% en 85% kan worden bereikt voor alle spelen. Dus ChanceProbCut vindt de beste zet sneller in expectimax zonder de speelsterkte te beïnvloeden. De uitgespaarde tijd kan worden geïnvesteerd in een diepere zoekboom. Experimenteel is gebleken dat er een kleine, maar relevante verbetering in speelsterkte is in Stratego en Dice. In ChanceBreakthrough werd een significante toename van de speelsterkte gemeten. ChanceProbCut wint 54,4% van 4.000 partijen.

In deterministische tweespeler spelen met perfecte informatie is het merendeel van het onderzoek gericht op de $\alpha\beta$ zoekmethode. Voor deterministische meerspeler spelen met perfecte informatie is de keuze van zoekmethode niet zo eenvoudig. De twee belangrijkste zoekmethoden zijn \max^n en paranoid. Ze benaderen beide het probleem vanuit een andere hoek. \max^n gaat ervan uit dat elke speler probeert de eigen score te maximaliseren, terwijl paranoid ervan uitgaat dat alle tegenstanders een coalitie gesloten hebben. Deze aannames hebben echter een nadeel. Vanwege het ontbreken van veilige snoeitechnieken kan \max^n slechts een beperkte zoekdiepte bereiken. Bovendien kan de onderliggende aanname van \max^n onrealistisch zijn. Dit kan resulteren in een te optimistisch spel. Bij het diepe zoeken met de paranoid zoekmethode kan de coalitie van andere spelers te dominant worden. Dit kan resulteren in een te pessimistisch spel. Dit heeft geleid tot de vierde onderzoeksvraag.

Onderzoeksvraag 4: *Hoe kunnen we het zoeken in meerspeler spelen verbeteren?*

Hoofdstuk 6 geeft antwoord op de vierde onderzoeksvraag door een nieuwe zoekmethode voor te stellen, genaamd Best-Reply Search (BRS), voor deterministische niet-coöperatieve meerspeler spelen met perfecte informatie. Dit algoritme laat slechts één tegenstander een zet spelen. Deze tegenstander is degene met de sterkste tegenzet. De andere spelers moeten passen. Met behulp van deze aanpak kunnen meer beurten van de wortelspeler worden doorzocht. Dit resulteert in een lange-termijnplanning. Tegelijkertijd wordt een soort van voorzichtigheid bewaard door rekening te houden met de sterkste tegenzet.

We hebben gekozen voor drie deterministische meerspeler spelen met perfecte informatie, Chinese Checkers, Focus, en Rolit. BRS is in staat om vergeleken met \max^n significant beter te presteren in deze spelen, en wint tussen de 65% en 95% van alle partijen. Tegen paranoid is BRS aanzienlijk sterker in Chinese Checkers en Focus, en wint tussen de 57% en 71% van alle partijen. In Rolit zijn BRS en paranoid gelijkwaardig. Bij het spelen tegen verschillende soorten tegenstanders tegelijkertijd is BRS de sterkste zoekmethode in Chinese Checkers en Focus. In Rolit is BRS enigszins zwakker dan paranoid. In het algemeen heeft het verhogen van de zoektijd geen negatief effect op de prestaties van BRS. Dit impliceert dat het zoeken van illegale posities, die worden gegenereerd door tegenstanders geforceerd te laten passen, geen groot invloed heeft. Een mogelijk negatief effect kan worden goedge maakt door een grotere zoekdiepte.

Hoofdstuk 7 geeft de conclusies van het proefschrift en geeft een vooruitblik op open vragen en aanwijzingen voor toekomstig onderzoek. Rekening houdend met de antwoorden op de bovenstaande onderzoeksvragen zien we dat er vier manieren zijn om selectieve zoekmethoden te verbeteren. Ten eerste, Single-Player Monte-Carlo Tree Search balanceert de exploitatie en exploratie zodanig dat het een waardig alternatief voor éénspeler spelen is waar geen goede evaluatiefunctie kan worden gevonden. Ten tweede, PN search met eindspeldatabases is in staat te bewijzen dat de speltheoretische waarde van het tweespeler spel Fanorona een remise is. Ten derde, ChanceProbCut maakt het mogelijk om takken van kansknopen voorwaarts

te snoeien in tweespeler spelen met niet-determinisme of imperfecte informatie. Ten vierde, in niet-coöperatieve deterministische meerspeler spelen met perfecte informatie realiseert Best-Reply Search langetermijnplanning door te veronderstellen dat het slechts één tegenstander toegestaan is om een tegenzet te spelen.

Hoewel dit proefschrift aantoont dat selectieve zoekmethoden succesvol zijn in veel domeinen, kunnen de door ons voorgestelde verbeteringen en zoekmethoden ook in andere domeinen worden getest. Deze domeinen omvatten klassen van spelen met hetzelfde niveau van complexiteit, maar ook klassen van spelen die niet zijn opgenomen in dit onderzoek. We noemen éenspeler spelen met niet-determinisme, tweespeler spelen met niet-determinisme en imperfecte informatie, en meerspeler spelen met niet-determinisme en/of imperfecte informatie.