

# Monte-Carlo Tree Search for Multi-Player Games

## Citation for published version (APA):

Nijssen, J. A. M. (2013). *Monte-Carlo Tree Search for Multi-Player Games*. [Doctoral Thesis, Maastricht University]. Maastricht University. <https://doi.org/10.26481/dis.20131202jn>

## Document status and date:

Published: 01/01/2013

## DOI:

[10.26481/dis.20131202jn](https://doi.org/10.26481/dis.20131202jn)

## 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](http://www.umlib.nl/taverne-license)

## Take down policy

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

[repository@maastrichtuniversity.nl](mailto:repository@maastrichtuniversity.nl)

providing details and we will investigate your claim.

# Summary

AI research has been quite successful in the field of two-player zero-sum games, such as chess, checkers, and Go. This has been achieved by developing two-player search techniques. However, a large number of games does not belong to the area where these search techniques are unconditionally applicable. Multi-player games are an example of such domains. This thesis focuses on two different categories of multi-player games: (1) deterministic multi-player games with perfect information and (2) multi-player hide-and-peek games. In particular, it investigates how Monte-Carlo Tree Search (MCTS) can be improved for games in these two categories. This technique has achieved impressive results in computer Go, but has also shown to be beneficial in a range of other domains.

Chapter 1 provides an introduction to games and the role they play in the field of AI and gives an overview of different game properties. It also defines the notion of multi-player games and discusses the two different categories of multi-player games that are investigated in this thesis. The following problem statement guides the research.

**Problem statement:** *How can Monte-Carlo Tree Search be improved to increase the performance in multi-player games?*

In order to answer the problem statement, four different research questions have been formulated. They deal with (1) incorporating different search policies in MCTS, (2) improving the selection phase of MCTS, (3) improving the playout phase of MCTS, and (4) adapting MCTS to a hide-and-peek game.

Chapter 2 introduces the basic definitions for game-tree search. It discusses the basic two-player minimax-based search techniques and some of their enhancements. Furthermore, this chapter describes three different minimax-based search techniques for multi-player games, namely  $\max^n$ , paranoid, and Best-Reply Search (BRS). Finally, it introduces MCTS, and how it can be applied to two-player and multi-player games.

Chapter 3 describes the test environment used to answer the first three research questions. We describe the rules and the employed domain knowledge for the four deterministic perfect-information games used in this thesis: Chinese Checkers, Focus, Rolit, and Blokus. Furthermore, for each of the games the state-space complexity and the game-tree complexity is given. They provide an indication on how difficult the games are for computers to play them optimally. Finally, this chapter introduces the

program MAGE, which is used to run the experiments in Chapters 4, 5, and 6 of this thesis.

The advantage of MCTS is that it can be extended to multi-player games. In the standard multi-player variant of MCTS, each player is concerned with maximizing his own win rate. This variant is therefore comparable to the minimax-based multi-player search technique  $\max^n$ , where each player tries to maximize his own score, regardless of the scores of the other players. Other multi-player search policies, such as the ones of paranoid and BRS, could also be considered. This led us to the first research question.

**Research question 1:** *How can multi-player search policies be incorporated in MCTS?*

Chapter 4 answers the first research question by incorporating the paranoid and BRS search policies, along with the default  $\max^n$  policy, in MCTS. With these search policies the selection and the backpropagation phases of MCTS are altered. In the MCTS framework, the  $\max^n$  search policy appeared to perform best. The advantages of paranoid and BRS in the minimax framework do not apply in MCTS, because  $\alpha\beta$  pruning is not applicable in MCTS. An additional problem with MCTS-BRS may be that, in the tree, invalid positions are investigated, which may reduce the reliability of the playouts as well. Still, MCTS-paranoid and MCTS-BRS overall achieved decent win rates against MCTS- $\max^n$ , especially with lower time settings. Based on the results we may conclude that the  $\max^n$  search policy is the most robust, though the BRS and paranoid search policies can still be competitive. Finally, we enhanced the  $\max^n$  search policy by proposing a multi-player variant of MCTS-Solver, called MP-MCTS-Solver. This variant is able to prove the game-theoretic value of a position. A win rate between 53% and 55% was achieved in the sudden-death game Focus. We may conclude that proving game-theoretic values improves the playing strength of MCTS in a multi-player sudden-death domain.

An important phase in the MCTS algorithm is the selection phase. During the selection phase, the search tree is traversed until a leaf node is reached. A selection strategy determines how the tree is traversed. Over the past years, several selection strategies and enhancements have been developed for different types of games. The most popular selection strategy is Upper Confidence bounds applied to Trees (UCT). There exist various enhancements for the UCT selection strategy. Most of them are domain dependent, which means that they cannot unconditionally be applied in every domain. A domain-independent enhancement is Rapid Action Value Estimation (RAVE). This enhancement is based on the all-moves-as-first (AMAF) heuristic. RAVE is in particular successful in the field of computer Go, but less successful in others, such as the multi-player game Chinese Checkers. This led us to the second research question.

**Research question 2:** *How can the selection phase of MCTS be enhanced in perfect-information multi-player games?*

Chapter 5 answers this question by proposing a new domain-independent selection strategy called Progressive History. This technique is a combination of the relative history heuristic and Progressive Bias. Contrary to RAVE, Progressive History

maintains its gathered data in a global table, in a similar way as the playout strategy Move-Average Sampling Technique. Progressive History was a significant improvement in all games with different numbers of players. In a comparison with UCT, Progressive History gained the highest win rates in the two-player variants, winning around 80% of the games. Moreover, Progressive History performed better than standard UCT in the multi-player variants as well. Progressive AMAF, which applies AMAF values instead of history values, overall performed significantly worse than Progressive History. Additionally, experiments in the two-player game Havannah showed that Progressive History performed better in this game than RAVE. Furthermore, experiments revealed that Progressive History also significantly increases the playing strength of the seekers in the hide-and-seek game Scotland Yard. Based on the results we may conclude that Progressive History considerably enhances MCTS in both two-player and multi-player games.

Similar to the selection phase, the playout phase is an important phase in the MCTS algorithm. During the playout phase, the game is finished by playing moves that are selected using a playout strategy. More realistic playouts usually provide more reliable results, thus increasing the playing strength of an MCTS-based player. Playouts can be made more realistic by adding domain knowledge. The disadvantage is that this may reduce the number of playouts per second, decreasing the playing strength. The challenge is to find a good balance between speed and quality of the playouts. For the two-player game Lines of Action (LOA), relatively time-expensive two-ply  $\alpha\beta$  searches in the playout phase of MCTS have been introduced. While this significantly reduced the number of playouts per second, it increased the overall playing strength by improving the quality of the playouts. This led us to the third research question.

**Research question 3:** *How can the playouts of MCTS be enhanced in perfect-information multi-player games?*

Chapter 6 answers this research question by introducing two-ply searches, which are equipped with a heuristic evaluation function, for selecting moves in the playout phase in MCTS for multi-player games. Three different search techniques were investigated for multi-player games, namely  $\max^n$ , paranoid and BRS. These playout strategies were compared against random, greedy and one-ply playouts to determine how to balance search and speed in the playouts of multi-player MCTS. The results showed that search-based playouts significantly improved the quality of the playouts in MCTS. Among the different playout strategies, BRS performed best, followed by paranoid and  $\max^n$ . This benefit was countered by a reduction of the number of playouts per second. Especially BRS and  $\max^n$  suffered from this effect. Among the tested two-ply search-based playouts, paranoid overall performed best with both short and long time settings. With more thinking time, the two-ply search-based playout strategies performed relatively better than the one-ply and greedy strategies. This indicates that with longer time settings, more computationally expensive playouts may be used to increase the playing strength of MCTS-based players. Based on the experimental results we may conclude that search-based playouts for multi-player games may be beneficial if the players receive sufficient thinking time.

The previous chapters discussed the application and enhancement of MCTS to deterministic multi-player games with perfect information. In Chapter 7, we shift our focus to hide-and-seek games. In this thesis we are interested in hide-and-seek games that have the following three properties. First, they feature imperfect information for some players. Second, some of the players have to cooperate in a fixed coalition. Though these players have a common goal, each player behaves autonomously and explicit communication between the players is not applied. Third, they are asymmetric. The different players have different types of goals. A game that features these properties is the pursuit-evasion game Scotland Yard. In this multi-player game, five seekers cooperate to try to capture a hider, which only shows its location on regular intervals. This led us to the fourth research question.

**Research question 4:** *How can MCTS be adapted for hide-and-seek games?*

Chapter 7 answers the fourth research question. For handling the imperfect information, two different determinization techniques were investigated, namely single-tree determinization and separate-tree determinization. Single-tree determinization had a slight overhead, but even when taking this into account, it performed significantly better than using separate trees. Furthermore, Location Categorization was proposed, which is a technique that can be used by both the MCTS and the expectimax seekers to give a better prediction for the location of the hider. It significantly increased the playing strength of both the MCTS and the expectimax seekers. The results gave empirical evidence that Location Categorization is a robust technique, as the weights worked for both seeker types against two different types of hider. Because of the asymmetric nature of the hide-and-seek game Scotland Yard, during the playouts, different playout strategies may be used by the different types of players. We found that, for the MCTS hider, it is best to assume during the playouts that the seekers do not know where the hider is, while the MCTS seekers perform best if they do assume where the hider is located. For dealing with the cooperation of the seekers, Coalition Reduction was proposed. This technique reduces the rewarded value for the root player if another player in the coalition wins the game, allowing the seekers to cooperate more effectively in the coalition. We observed that the performance of the MCTS seekers increased by applying Coalition Reduction. Cooperation still appeared to be important, because the performance of the seekers dropped significantly when the reduction became too large. In a direct comparison, MCTS performed considerably better than paranoid search for the hider and expectimax for the seekers. Finally, the experimental results showed that MCTS was able to play Scotland Yard on a higher level than a commercial Nintendo DS program, which is generally considered to be a strong player. In conclusion, with the incorporation of enhancements such as single-tree determinization, Location Categorization, and Coalition Reduction, we were able to let an MCTS-based player play the hide-and-seek game Scotland Yard on a strong level.

Chapter 8 concludes the thesis and provides an outlook on five directions for future research. The answer to the problem statement may be summarized in four points, based on the research questions. First, the  $\max^n$  search policy performs the best in multi-player MCTS, while the BRS and paranoid policies are still competitive. The

$\max^n$  search policy can be enhanced with a multi-player variant of the MCTS-Solver. Second, the Progressive History selection strategy significantly increases the performance of two-player and multi-player MCTS. Third, two-ply search-based playouts significantly improve the quality of the playouts and, assuming a sufficient amount of thinking time is provided, increases the performance of MCTS in multi-player domains. Fourth, incorporating single-tree determinization, Location Categorization, and Coalition Reduction into MCTS significantly improves its performance in the multi-player hide-and-seek game Scotland Yard.

The research presented in this thesis indicates five areas of future research. These areas include (1) the application of other search policies, (2) the combination of Progressive History with other selection strategies, (3) further optimization of search-based playouts or the implementation of three-ply search-based playouts, (4) further investigation of Scotland Yard, and (5) the application if the proposed enhancements to other domains.



# Samenvatting

Onderzoek in kunstmatige intelligentie is succesvol geweest op het gebied van tweespeler nulsomspelen, zoals schaak, dammen en Go. Dit is bereikt door het ontwikkelen van tweespeler zoektechnieken. Echter, een groot aantal spelen behoort tot de groep waar deze technieken niet zonder meer kunnen worden toegepast. Meerspeler spelen zijn een voorbeeld van zo'n domein. Dit proefschrift richt zich op twee verschillende categorieën meerspeler spelen: (1) deterministische meerspeler spelen met perfecte informatie en (2) meerspeler zoek-en-verstopspelen. In het bijzonder wordt onderzocht hoe Monte-Carlo Tree Search (MCTS) voor spelen in deze twee categorieën kan worden verbeterd. Deze techniek heeft indrukwekkende resultaten behaald in computer Go, maar is ook waardevol gebleken in een reeks andere domeinen. Hoofdstuk 1 biedt een introductie in spelen en de rol die zij vervullen op het gebied van kunstmatige intelligentie, en daarnaast geeft het een overzicht van verschillende speleigenschappen. Ook worden meerspeler spelen gedefinieerd en worden de twee verschillende categorieën meerspeler spelen die onderzocht worden in dit proefschrift besproken. De volgende probleemstelling stuurt het onderzoek.

**Probleemstelling:** *Hoe kan Monte-Carlo Tree Search verbeterd worden om de prestaties in meerspeler spelen te verhogen?*

Om een antwoord te geven op de probleemstelling, worden vier onderzoeksvragen gedefinieerd. Deze behandelen (1) de integratie van verschillende zoekprincipes in MCTS, (2) het verbeteren van de selectiefase van MCTS, (3) het verbeteren van de simulatiefase van MCTS, en (4) het aanpassen van MCTS voor zoek-en-verstopspelen.

Hoofdstuk 2 introduceert de basisdefinities voor het zoeken in spelbomen. De fundamentele tweespeler minimax-gebaseerde zoektechnieken en enkele verbeteringen hierop worden besproken. Verder beschrijft dit hoofdstuk drie minimax-gebaseerde zoektechnieken voor meerspeler spelen, namelijk  $\max^n$ , paranoid en Best-Reply Search (BRS). Tenslotte wordt MCTS geïntroduceerd, en wordt er beschreven hoe het kan worden toegepast in meerspeler spelen.

Hoofdstuk 3 beschrijft de testomgeving die gebruikt wordt om de eerste drie onderzoeksvragen te beantwoorden. We beschrijven de regels en de gebruikte domeinkennis voor de vier deterministische perfecte-informatie spelen die ingezet worden in dit proefschrift: Chinese Checkers, Focus, Rolit en Blokus. Vervolgens wordt voor ieder van deze spelen de complexiteit van de toestandsruimte en van de spelboom gegeven. Deze geven een indicatie van hoe moeilijk de spelen zijn voor computers om ze



optimaal te spelen. Tenslotte introduceert dit hoofdstuk het programma MAGE, dat gebruikt wordt om de experimenten in hoofdstukken 4, 5 en 6 uit te voeren.

Het voordeel van MCTS is dat het kan worden uitgebreid naar meerspeler spelen. In de standaard variant van MCTS is iedere speler enkel bezig met het maximaliseren van zijn eigen winstratio. Deze variant is daardoor vergelijkbaar met de minimax-gebaseerde meerspeler zoektechniek  $\max^n$ , waar iedere speler probeert zijn eigen score te maximaliseren, ongeacht de scores van de andere spelers. Andere meerspeler zoekprincipes, zoals die van paranoid en BRS, kunnen ook worden bekeken. Dit heeft geleid tot de eerste onderzoeksvraag.

**Onderzoeksvraag 1:** *Hoe kunnen meerspeler zoekprincipes worden gebruikt in MCTS?*

Hoofdstuk 4 beantwoordt de eerste onderzoeksvraag door de paranoid en BRS zoekprincipes, samen met het standaard  $\max^n$  principe, te gebruiken in MCTS. Middels deze drie principes worden de selectie- en de terugpropagatiefase van MCTS aangepast. In het MCTS raamwerk lijkt het  $\max^n$  zoekprincipe het beste te werken. De voordelen van paranoid en BRS in het minimax raamwerk zijn niet van toepassing in MCTS, omdat  $\alpha\beta$  snoeiing niet toepasbaar in MCTS is. Een bijkomend probleem van MCTS-BRS is dat in de boom onjuiste posities worden onderzocht, wat de betrouwbaarheid van de simulaties kan verlagen. Echter, MCTS-paranoid en MCTS-BRS behalen over het algemeen redelijke winstpercentages tegen MCTS- $\max^n$ , zeker bij lagere tijdsinstellingen. Op basis van de resultaten mogen we concluderen dat het  $\max^n$  zoekprincipe het meest robuust is, hoewel de BRS en paranoid zoekprincipes nog steeds competitief zijn. Tenslotte verbeterden we het  $\max^n$  zoekprincipe met de meerspeler variant van de MCTS-Solver, genaamd MP-MCTS-Solver. Deze variant is in staat om speltheoretische waarden in een positie te bewijzen. Een winstpercentage tussen 53% en 55% werd bereikt in het sudden-death spel Focus. We mogen concluderen dat het bewijzen van de speltheoretische waarden de speelsterkte van MCTS in een meerspeler sudden-death domein verbetert.

Een belangrijke fase in het MCTS algoritme is de selectiefase. Tijdens deze fase wordt de boom doorlopen totdat een blad is bereikt. Een selectiestrategie bepaalt hoe de boom wordt doorlopen. Gedurende de afgelopen jaren zijn verschillende selectiestrategieën en verbeteringen ontwikkeld voor verschillende soorten spelen. De populairste selectiestrategie is Upper Confidence bounds applied to Trees (UCT). Er bestaan verscheidene verbeteringen voor de UCT selectiestrategie. De meesten zijn domeinafhankelijk, wat betekent dat ze niet zonder meer in ieder domein toegepast kunnen worden. Een domeinonafhankelijke verbetering is Rapid Action Value Estimation (RAVE). Deze verbetering is gebaseerd op de all-moves-as-first (AMAF) heuristiek. RAVE is behoorlijk succesvol in het gebied van computer Go, maar minder in andere gebieden, zoals het meerspeler spel Chinese Checkers. Dit heeft geleid tot de tweede onderzoeksvraag.

**Onderzoeksvraag 2:** *Hoe kan de selectiefase van MCTS worden verbeterd in perfecte-informatie meerspeler spelen?*

Hoofdstuk 5 beantwoordt deze vraag door het voorstellen van een nieuwe domeinonafhankelijke selectiestrategie genaamd Progressive History. Deze techniek is een

combinatie van de relatieve historie-heuristiek en Progressive Bias. In tegenstelling tot RAVE slaat Progressive History de verzamelde data op in een globale tabel, op een manier die vergelijkbaar is met de simulatiestrategie Move-Average Sampling Technique. Progressive History bleek een significante verbetering in alle spelen met verschillende aantallen spelers. In een vergelijking met UCT behaalde Progressive History de hoogste winstpercentages in de tweespeler varianten, waar het ongeveer 80% van de partijen wint. Bovendien presteert Progressive History ook in de meerspeler varianten beter dan UCT. Progressive AMAF, dat AMAF waarden gebruikt in plaats van history waarden, presteert over het algemeen slechter dan Progressive History. Experimenten in het tweespeler spel Havannah lieten zien dat Progressive History in dit spel beter presteert dan RAVE. Daarnaast bleek ook dat Progressive History de prestatie van de zoekers in het zoek-en-verstopspel Scotland Yard significant verbetert. Op basis van de resultaten mogen we concluderen dat Progressive History MCTS aanzienlijk verbetert in zowel tweespeler als in meerspeler spelen.

Net als de selectiefase is de simulatiefase een belangrijk onderdeel in het MCTS algoritme. Tijdens deze fase wordt de partij voltooid met zetten die worden geselecteerd middels een simulatiestrategie. Realistischere simulaties bieden betrouwbaardere resultaten, wat de speelsterkte van een MCTS-gebaseerde speler verbetert. Simulaties kunnen realistischer gemaakt worden door het toevoegen van domeinkennis. Het nadeel is dat dit het aantal simulaties per seconde kan verkleinen, hetgeen de speelsterkte verlaagt. De uitdaging is om een goede balans te vinden tussen snelheid en kwaliteit van de simulaties. Voor het tweespeler spel Lines of Action (LOA) zijn relatief dure 2-ply  $\alpha\beta$  zoekprocessen geïntroduceerd. Hoewel deze het aantal simulaties per seconde significant verkleinden, nam de speelsterkte door het verbeteren van de kwaliteit van de simulaties toe. Dit heeft geleid tot de derde onderzoeksvraag.

**Onderzoeksvraag 3:** *Hoe kunnen de simulaties van MCTS verbeterd worden in perfecte-informatie meerspeler spelen?*

Hoofdstuk 6 beantwoordt deze vraag door het introduceren van 2-ply zoekprocessen, uitgerust met een heuristische evaluatiefunctie, die gebruikt worden voor het selecteren van zetten in de simulatiefase van MCTS voor meerspeler spelen. Drie verschillende technieken voor meerspeler spelen zijn onderzocht, namelijk  $\max^n$ , paranoid, en BRS. Deze simulatiestrategieën werden vergeleken met willekeurig, greedy, en 1-ply simulaties om te bepalen hoe exploratie en snelheid gebalanceerd dienen te worden in de simulaties van meerspeler MCTS. De resultaten lieten zien dat zoekgebaseerde simulaties de kwaliteit van de simulaties significant verhoogden. Onder de verschillende simulatiestrategieën presteerde BRS het beste, gevolgd door paranoid en  $\max^n$ . Dit voordeel werd afgezwakt door het lagere aantal simulaties per seconde. Vooral BRS en  $\max^n$  leden hieronder. Onder de geteste 2-ply zoekgebaseerde simulaties presteerde paranoid over het algemeen het beste met zowel korte als lange tijdsinstellingen. Met meer denktijd presteerden de 2-ply zoekgebaseerde simulatiestrategieën relatief beter dan de 1-ply en greedy strategieën. Dit geeft aan dat met langere tijdsinstellingen de computationeel duurdere simulaties gebruikt kunnen worden om de speelsterkte van MCTS-gebaseerde spelers te verhogen. Op basis

van de experimentele resultaten mogen we concluderen dat zoekgebaseerde simulaties voor meerspeler spelen voordelig kunnen zijn als de spelers voldoende denktijd krijgen.

De voorgaande hoofdstukken bespraken de toepassing en verbetering van MCTS in deterministische meerspeler spelen met perfecte informatie. In hoofdstuk 7 verschuiven we onze aandacht naar zoek-en-verstopspelen. In dit proefschrift zijn we geïnteresseerd in zoek-en-verstopspelen met de volgende drie eigenschappen. Ten eerste bevatten ze imperfecte informatie voor sommige spelers. Ten tweede spelen sommige spelers samen in een vaste coalitie. Ten derde zijn ze asymmetrisch. De verschillende spelers hebben verschillende soorten doelen. Een spel dat deze eigenschappen heeft is het spel Scotland Yard. In dit meerspeler spel werken vijf zoekers samen om te proberen een verstopper te vangen, die zijn locatie slechts op reguliere tussenpozen bekendmaakt. Dit heeft geleid tot de vierde onderzoeksvraag.

**Onderzoeksvraag 4:** *Hoe kan MCTS aangepast worden voor een zoek-en-verstopspel?*

Hoofdstuk 7 beantwoordt de vierde onderzoeksvraag. Voor het omgaan met de imperfecte informatie werden twee verschillende determinisatietechnieken onderzocht, namelijk enkele-boom determinisatie en aparte-boom determinisatie. Enkele-boom determinisatie heeft een kleine overhead, maar zelfs als we hier rekening mee houden presteert het significant beter dan het gebruik van aparte bomen. Vervolgens werd Location Categorization voorgesteld, wat een techniek is dat gebruikt kan worden door zowel de MCTS als de expectimax zoekers om een betere voorspelling te maken van de locatie van de verstopper. Het verhoogde de speelsterkte van zowel de MCTS als de expectimax zoekers significant. De resultaten gaven empirisch bewijs dat Location Categorization een robuuste techniek is, omdat de gewichten werkten voor twee verschillende soorten zoekers tegen twee verschillende soorten verstoppers. Vanwege de asymmetrische natuur van Scotland Yard kunnen verschillende simulatiestrategieën voor de verschillende soorten spelers gebruikt worden in de simulaties. We zagen dat het voor de MCTS verstopper beter was als hij tijdens de simulaties aanneemt dat de zoekers niet weten waar hij zit, terwijl de MCTS zoekers het beste presteren als ze aannemen dat ze weten waar de verstopper zit. Om de coalitie tussen de zoekers te behandelen werd Coalition Reduction voorgesteld. Deze techniek verlaagt de toegekende waarde voor de wortelspeler als een andere speler in de coalitie de partij wint, wat de zoekers in staat stelt efficiënter aan de coalitie deel te nemen. We zagen dat de prestatie van de MCTS zoekers verbeterde door het toepassen van Coalition Reduction. Samenwerking bleek nog steeds belangrijk, want de prestatie van de zoekers daalde significant als de reductie te groot werd. In een directe vergelijking bleek MCTS aanzienlijk beter te presteren dan het paranoid zoekproces voor de verstopper en expectimax voor de zoekers. Tenslotte lieten de experimentele resultaten zien dat MCTS in staat was om op een hoger niveau te spelen dan het commercieel Nintendo DS programma, dat over het algemeen wordt gezien als een sterke speler. Dus we mogen concluderen dat we met de toevoeging van verbeteringen als enkele-boom determinisatie, Location Categorization, en Coalition Reduction in staat zijn om een MCTS-gebaseerde speler het zoek-en-verstopspel Scotland Yard op een sterk niveau te laten spelen.

Hoofdstuk 8 sluit het proefschrift af en geeft een vooruitblik op vijf richtingen voor toekomstig onderzoek. Het antwoord op de probleemstelling kan in vier punten worden samengevat. Ten eerste presteert het  $\max^n$  zoekprincipe het beste in meerspeler MCTS, terwijl BRS en paranoid principes nog steeds competitief kunnen zijn. Het  $\max^n$  zoekprincipe kan worden verbeterd met een meerspeler variant van MCTS-Solver. Ten tweede verbetert de Progressive History selectiestrategie de prestatie van tweespeler en meerspeler MCTS significant. Ten derde, 2-ply zoekgebaseerde simulaties verbeteren de kwaliteit van de simulaties significant en, aangenomen dat voldoende denktijd beschikbaar is, verbetert de prestatie van MCTS in meerspeler domeinen. Ten vierde, de toevoeging van enkele-boom determinisatie, Location Categorization, en Coalition Reduction aan MCTS verbeteren de prestaties ervan significant in het meerspeler zoek-en-verstopspel Scotland Yard.

Het onderzoek dat gepresenteerd is in dit proefschrift duidt vijf gebieden voor toekomstig onderzoek aan. Deze gebieden bevatten (1) de toepassing van andere zoekprincipes, (2) de combinatie van Progressive History met andere selectiestrategieën, (3) het verder optimaliseren van de implementatie van 3-ply zoekgebaseerde simulaties, (4) het verder onderzoeken van Scotland Yard, en (5) de toepassing van de voorgestelde verbeteringen in andere domeinen.