Challenging Established Move Ordering Strategies with Adaptive Data Structures

Spencer Polk and B. John Oommen

School of Computer Science, Carleton University, Ottawa, Canada
andrewpolk@cmail.carleton.ca, oommen@scs.carleton.ca**

Abstract. The field of game playing is a particularly well-studied area within the context of AI, leading to the development of powerful techniques, such as the alpha-beta search, capable of achieving competitive game play against an intelligent opponent. It is well known that tree pruning strategies, such as alpha-beta, benefit strongly from proper move ordering, that is, searching the best element first. Inspired by the formerly unrelated field of Adaptive Data Structures (ADSs), we have previously introduced the History-ADS technique, which employs an adaptive list to achieve effective and dynamic move ordering, in a domain independent fashion, and found that it performs well in a wide range of cases. However, previous work did not compare the performance of the History-ADS heuristic to any established move ordering strategy. In an attempt to address this problem, we present here a comparison to two well-known, acclaimed strategies, which operate on a similar philosophy to the History-ADS, the History Heuristic, and the Killer Moves technique. We find that, in a wide range of two-player and multi-player games, at various points in the game's progression, the History-ADS performs at least as well as these strategies, and, in fact, outperforms them in the majority of cases.

1 Introduction

Achieving competitive play in a strategic board game, against one or more intelligent opponents, is a canonical problem within the field of AI. From the inception of the field to the present, a broad corpus of literature has been published on this topic, introducing a wide range of strategies to achieve effective game play, in a wide range of board games [1–3]. In particular, one of the most studied and acclaimed techniques is the alpha-beta search, which is capable of achieving a much greater look-ahead, or search depth, in game trees, by pruning large sections of the search space [4, 5]. It is furthermore well-established that the efficiency of alpha-beta pruning is highly dependent on proper move ordering, that is, searching the strongest moves at each level of the tree first, and a range of move ordering heuristics have been developed to achieve this [2, 6].

** *Chancellor's Professor*; *Fellow: IEEE* and *Fellow: IAPR*. The second author is also an *Adjunct Professor* with the Dept. of ICT, University of Agder, Grimstad, Norway.

These include the highly regarded and well-studied History Heuristic, and the Killer Moves strategy [6, 7].

Although a broad range of techniques for achieving competitive game play have been introduced, the majority of the literature focuses on Two-Player games, such as Chess and Go, with substantially less emphasis placed on Multi-Player (MP) games [2, 8]. Indeed, it is well known that most of the techniques for MP games are extensions of corresponding two-player strategies, and often have trouble performing on the level of their counterparts, for a variety of reasons [9–12]. In an attempt to improve MP strategies, in earlier papers, we derived techniques from the formerly unrelated field of Adaptive Data Structures (ADSs), a field concerned with reorganizing a data structure dynamically, to match query frequencies [13, 14]. A technique to improve move ordering in a state-of-the-art MP algorithm, the Threat-ADS heuristic, was introduced in [15], and expanded on in [16][1]. Based on this success, we later generalized ADS-based strategies to both two-player and MP environments, introducing the History-ADS heuristic [17]. The History-ADS heuristic operates by ranking potential moves, based on their previous performance within the tree, using a list-based ADS.

The History-ADS heuristic demonstrated an ability to produce substantial improvements in terms of tree pruning in a wide range of cases, and without a substantial investment in terms of computational resources [17, 18]. However, while the History-ADS heuristic has produced known benefits, it has not been directly compared to known previously-reported move ordering strategies. Given its conceptual similarities to the History Heuristic and the Killer Moves strategy, in this work we present a comparison of its performance to these two well-known, highly regarded techniques, and demonstrate that it is capable of performing on their level, and in fact, outperforming them in some cases.

The rest of the paper is laid out as follows. Section 2 discusses in detail the motivation behind our work in this paper, and Section 3 describes the Threat-ADS and History-ADS techniques in depth. Section 4 describes our experimental design, and the game models we will be employing in our work. Section 5 presents our results for both two-player and MP games, and Section 6 provides our discussion and analysis of these results. Lastly, Section 7 concludes the paper.

2 Motivation

Our previous work in [17] and [18] demonstrated the potential benefits of the History-ADS heuristic in a wide range of environments, and explored a large number of possible configurations within these environments. Unlike the Threat-ADS heuristic, however, which pioneered an entirely new concept, i.e., that of using opponent threats to achieve move ordering, the History-ADS heuristic achieves move ordering through move history, a known metric. While large reductions, of over 75%, for a relatively lightweight technique, clearly demonstrate the success of the History-ADS, we cannot be sure of its actual relative benefits, unless it is compared to established, well-known techniques of a similar nature.

[1] The latter paper won the *Best Paper Award* of the IEA/AIE conference in 2015.

Given the similarities in principle behind the History-ADS heuristic, and both the History Heuristic and Killer Moves, we have elected to present a comparison between its performance and these two well-known methods, under a similar testing domain to that employed in our earlier work. By providing this comparison, we believe that we will be able to place the History-ADS heuristic's achievements in the proper context and perspective. Furthermore, it may actually be the case that the History-ADS heuristic is able to outperform one or both of them, which would be a very valuable result, given its inexpensive cost.

In fact, we have reason to suspect that the History-ADS heuristic may be capable of outperforming the History Heuristic in at least some domains. This is based on the premise of the results presented in [18]. The History Heuristic employs a relatively complex mechanism to rank moves, compared to the History-ADS heuristic. It is, however, less sensitive to change than the History-ADS heuristic, when employing a "Move-to-Front" adaptive list, which was found to perform better than strategies that are more conservative in their structural changes in our previous work. It may prove to be the case that in at least some domains, the extreme adaptability of the Move-to-Front rule will outperform even the elaborate History Heuristic. This begs investigation.

The Killer Moves heuristic is already conceptually very similar to the History-ADS operating with a "multi-level" ADS, introduced in [18] (this concept will be described in detail in the next section). In that paper, we found that a single ADS generally outperformed the multi-level variant, despite some potential drawbacks of applying the same list at all levels of the tree. Given that the Killer Moves strategy typically retains fewer moves, at each level of the tree, than any of the multi-level approaches we have previously explored, we highly suspect that the History-ADS, employing a single list, will be able to outperform it.

The potential to achieving performance on the level of the well-regarded History Heuristic and Killer Moves, using an ADS-based strategy, motivates the work presented in this paper.

3 Previous Work

ADSs were, as mentioned earlier, were originally designed to reorganize their structure, in response to queries over time, to better match access frequencies [13, 14]. An example of a specific ADS *update mechanism*, for adaptive lists, is the Move-to-Front rule, where the accessed element is moved to the head of the list, and thus, will tend to remain close to the front if it is frequently accessed. The reader will observe, however, that this organization also provides an intuitive mechanism by which the elements of the data structure could be ranked. Our previous work is based on harnessing the lightweight mechanics of ADSs to serve as a ranking mechanism for elements of a game, such as players, moves, or board positions, and leveraging this ranking to achieve improvements in performance. Currently, we have focused on achieving better move ordering, and thus tree pruning, although this strategy may have broader applications.

Threat-ADS: Our first attempt to apply ADS-based techniques to game playing, was motivated by the desire to improve performance in the under-studied field of MP games. In a MP environment, as opposed to the two-player case, there are many opponents to consider, instead of just one. Intuitively, each of these opponents may threaten the player to a different extent. We focused on the Best-Reply Search (BRS), a recent, powerful MP strategy, which seeks to manage the complex case of multiple opponents by unifying them into a single "super-opponent" in its search, which minimizes the player [10]. We observed that the BRS did not unify the moves for each opponent, at each MIN level of the tree, in a specific manner, and thus introduced the Threat-ADS, which uses a small adaptive list containing opponents, to dynamically determine the best method to do this. An example of the Threat-ADS heuristic in action is shown in Figure 1. The Threat-ADS heuristic was found to produce statistically significant results in a wide range of cases, considering different update mechanisms, ply depths, and games [16, 19].

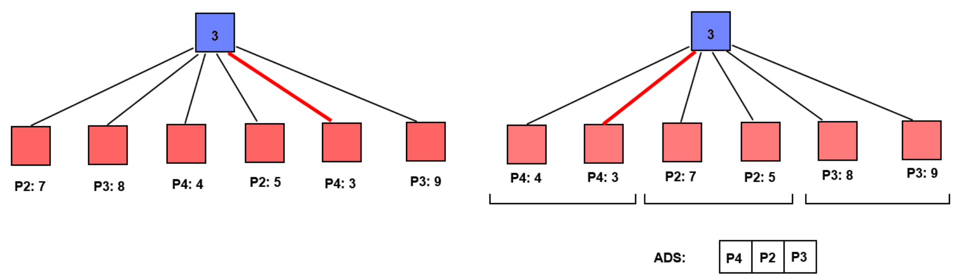

Fig. 1. The BRS without Threat-ADS on the left, and with it on the right. Notice how a cut is made faster in the second case.

History-ADS: Based on the success of the ADS-based Threat-ADS heuristic, we sought to generalize its specific, MP approach, to be applicable to both MP and two-player games. Rather than seeking to rank opponents, which only has applicability in the MP space, we drew inspiration from the well-known History Heuristic, and employed a list-based ADS to rank moves. The History-ADS heuristic operates in the context of the alpha-beta search, maintaining a list of possible moves. When a move produces a cut, the ADS is "queried" with the identity of that move, and it is moved towards the head of the list, according to the ADS' update mechanism. When moves are explored at a new level of the tree, this is done in the order dictated by the ADS, if applicable, similar to exploring the killer moves first. An example of the History-ADS heuristic in action is provided in Figure 2.

Results from previous work demonstrated clearly that the History-ADS heuristic was capable of obtaining very large reductions in the tree size, through improved pruning, in a wide range of cases [17]. It was furthermore shown to perform best using the Move-to-Front update mechanism, relative to less sen-

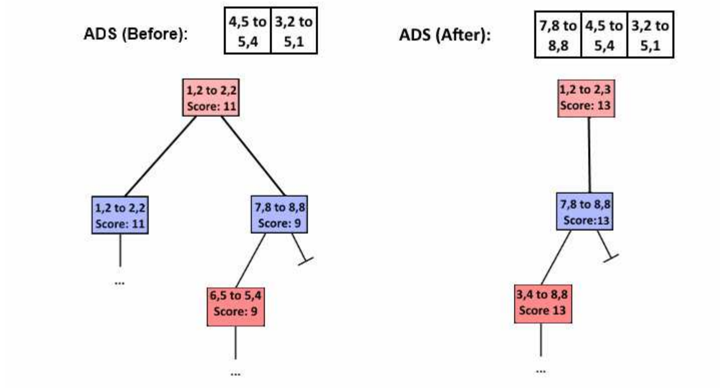

Fig. 2. A demonstration of how an ADS can be used to manage move history over time. The move (7,8) to (8,8) produces a cut, and so it is moved to the head of the list, and informs the search later.

sitive strategies, and that it could keep most of its performance with a strong restriction on the length of the list [18].

4 Game Models and Experimental Setup

Given that our work in this paper is a logical "next step" of our previously published work, it is sensible for us to employ an analogous set of experiments, and employ a similar software framework, to that which was employed in [17]. We are interested in the improvement in tree pruning, when employing the History-ADS heuristic, relative to the established History Heuristic and Killer Moves techniques. We accomplish this measurement by recording an aggregate of the Node Count (NC) over several turns of the game. The NC is defined as the number of nodes that are expanded during the search, i.e. excluding those generated but then pruned before being visited. Historically, this metric has been shown to be highly correlated to runtime, while also being platform-agnostic [6]. For a variety of MP and two-player games, we average this value over fifty trials, for each technique in question.

As in our previous work, we will employ the known MP games Focus, and Chinese Checkers, as well as a territory control game of our own devising, which we have named the Virus Game, the rules for which are described in [15]. We will employ the same two-player games as well, including the two-player variant of Focus, the game Othello, and the very well-known Checkers, or Draughts. However, the requirement in Checkers that forces jumps when possible, often leads to a game with a very small branching factor, with a highly variable number of moves available at the midgame state. Thus, for our experiments, we choose to relax this rule, and do not require that a player must necessarily make an available jump. We shall refer to this game as "Relaxed Checkers". While Checkers has been solved, nevertheless serves as a useful testing environment for the general applicability of a domain-independent strategy, given how well-known and documented the game is in the literature [20].

Rather than simply test the performance of the History-ADS and other techniques from the initial board state, which can be relatively unchanging between games, we also provide results for the midgame case, as we did previously in [16]. In order to generate reasonable midgame states, we have intelligent agents play the game for a number of turns, before measurements take place, in each trial. The details of how midgame states are generated is described in greater detail in [16]. The number of turns the games are advanced for has been refined through observation, and the values we use in this work are 15 for the Virus Game, 5 for Focus (both MP and two-player versions) and Relaxed Checkers, and 10 for Othello and Chinese Checkers.

The number of turns we aggregate the NC over, for both initial and midgame cases, is 5 for Relaxed Checkers, Othello, and Chinese Checkers, 3 for Focus (given its short duration), and 10 for the Virus Game. To determine statistical significance, we employ the Mann-Whitney test due to a lack of guaranteed normalcy in the data. Lastly, we provide the Effect Sizes, which serve as an easily readable indication of the degree of the savings in terms of tree pruning. We employ a domain-independent version of the History Heuristic, based on its original specification in [6].

Our results are presented in the following section.

5 Results

The following sections present our results, as well as our statistical analysis, of the History-ADS heuristic in comparison to the History Heuristic, and the Killer Moves technique, in both two-player and multi-player contexts.

5.1 Results for Two-Player Games

Our results for Othello are presented in Table 1. We observe that in both the initial board position and midgame cases, the History-ADS heuristic outperformed both the History Heuristic, and Killer Moves, which behaved very similarly to each other. For example, in the midgame case, the History-ADS represented a 14% improvement over the History Heuristic and Killer Moves techniques.

Table 2 showcases our results for Relaxed Checkers. Again, the History-ADS performed best in all situations, however the Killer Moves technique outperformed the History Heuristic, in this game. The History-ADS did 25% better than the History Heuristic, when measurements were taken from the initial board position.

In Table 3, we present our results for two-player focus. As is consistent with previous work, all techniques led to a drastic reduction in NC, although again, History-ADS did best of all, and Killer Moves outperformed the History Heuristic.

Table 1. Results comparing the History-ADS, History Heuristic, and Killer Moves in Othello.

Midgame?	Technique	Avg. NC	Std. Dev	P-Value	Effect Size
No	None	5,061	2,385	-	-
No	History-ADS	3,727	1,552	1.7×10^{-3}	0.56
No	History Heuristic	4,136	1,711	0.071	0.37
No	Killer Moves	4,013	1,720	0.015	0.44
Yes	None	20,100	9,899	-	-
Yes	History-ADS	13,300	6,916	7.0×10^{-5}	0.69
Yes	History Heuristic	15,500	6,939	9.4×10^{-3}	0.47
Yes	Killer Moves	15,500	6,696	0.015	0.47

Table 2. Results comparing the History-ADS, History Heuristic, and Killer Moves in Relaxed Checkers.

Midgame?	Technique	Avg. NC	Std. Dev	P-Value	Effect Size
No	None	78,600	10,600	-	-
No	History-ADS	41,000	5,588	$< 1.0 \times 10^{-5}$	3.55
No	History Heuristic	54,800	9,018	$< 1.0 \times 10^{-5}$	2.25
No	Killer Moves	52,200	6,723	$< 1.0 \times 10^{-5}$	2.50
Yes	None	64,000	25,700	-	-
Yes	History-ADS	34,400	12,400	$< 1.0 \times 10^{-5}$	1.15
Yes	History Heuristic	42,800	14,500	$< 1.0 \times 10^{-5}$	0.83
Yes	Killer Moves	39,100	14,700	$< 1.0 \times 10^{-5}$	0.97

5.2 Results for Multi-Player Games

Table 4 holds our results for the Virus Game, where the same patterns as previously were observed. The History Heuristic performed particularly poorly here, with History-ADS representing 28% improvement over it in the midgame situation.

In Table 5, we show our results for MP Focus, which again, follow the established pattern. In this case, the difference between History-ADS and Killer Moves was negligible, although History-ADS outperformed it in both cases.

Finally, Table 6 presents our results for Chinese Checkers. We see the pattern observed with the other games repeated again, although Killer Moves was very close to History-ADS in this case, being approximately equivalent in the

Table 3. Results comparing the History-ADS, History Heuristic, and Killer Moves in Two-Player Focus.

Midgame?	Technique	Avg. NC	Std. Dev	P-Value	Effect Size
No	None	5,250,000	381,000	-	-
No	History-ADS	1,260,000	90,900	$< 1.0 \times 10^{-5}$	10.46
No	History Heuristic	1,980,000	221,000	$< 1.0 \times 10^{-5}$	8.59
No	Killer Moves	1,420,000	105,100	$< 1.0 \times 10^{-5}$	10.04
Yes	None	10,600,000	3,460,000	-	-
Yes	History-ADS	2,390,000	631,000	$< 1.0 \times 10^{-5}$	2.37
Yes	History Heuristic	3,500,000	1,040,000	$< 1.0 \times 10^{-5}$	2.05
Yes	Killer Moves	2,680,000	648,000	$< 1.0 \times 10^{-5}$	2.29

Table 4. Results comparing the History-ADS, History Heuristic, and Killer Moves in the Virus Game.

Midgame?	Technique	Avg. NC	Std. Dev	P-Value	Effect Size
No	None	10,500,000	1,260,000	-	-
No	History-ADS	4,650,000	767,000	$< 1.0 \times 10^{-5}$	4.60
No	History Heuristic	6,860,000	1,080,000	$< 1.0 \times 10^{-5}$	2.86
No	Killer Moves	5,210,000	858,000	$< 1.0 \times 10^{-5}$	4.16
Yes	None	12,800,000	1,950,000	-	-
Yes	History-ADS	5,870,000	863,000	$< 1.0 \times 10^{-5}$	3.55
Yes	History Heuristic	8,190,000	1,080,000	$< 1.0 \times 10^{-5}$	2.36
Yes	Killer Moves	6,380,000	991,000	$< 1.0 \times 10^{-5}$	3.29

midgame case, and the History Heuristic did noticeably worse. In the midgame case, the History-ADS heuristic did 37% better than the History Heuristic.

6 Discussion

Our results clearly demonstrate the power of the History-ADS heuristic, even when compared to the established, highly-regarded techniques, specifically, the Killer Moves strategy, and the History Heuristic. We found that in nearly every case examined, the History-ADS heuristic outperformed both of these established techniques, or performed on a level comparable to them.

Although we were somewhat surprised that the History Heuristic was outperformed by the History-ADS heuristic in every case examined, reviewing our

Table 5. Results comparing the History-ADS, History Heuristic, and Killer Moves in Multi-Player Focus.

Midgame?	Technique	Avg. NC	Std. Dev	P-Value	Effect Size
No	None	6,970,000	981,000	-	-
No	History-ADS	2,150,000	165,000	$< 1.0 \times 10^{-5}$	4.92
No	History Heuristic	3,360,000	351,000	$< 1.0 \times 10^{-5}$	3.69
No	Killer Moves	2,220,000	175,000	$< 1.0 \times 10^{-5}$	4.84
Yes	None	14,200,000	8,400,000	-	-
Yes	History-ADS	3,160,000	1,700,000	$< 1.0 \times 10^{-5}$	1.31
Yes	History Heuristic	5,050,000	3,010,000	$< 1.0 \times 10^{-5}$	1.09
Yes	Killer Moves	3,260,000	1,530,000	$< 1.0 \times 10^{-5}$	1.30

Table 6. Results comparing the History-ADS, History Heuristic, and Killer Moves in Chinese Checkers.

Midgame?	Technique	Avg. NC	Std. Dev	P-Value	Effect Size
No	None	3,370,000	1,100,000	-	-
No	History-ADS	1,280,000	368,000	$< 1.0 \times 10^{-5}$	1.90
No	History Heuristic	1,550,000	445,000	$< 1.0 \times 10^{-5}$	1.66
No	Killer Moves	1,310,000	341,000	$< 1.0 \times 10^{-5}$	1.88
Yes	None	8,260,000	1,950,000	-	-
Yes	History-ADS	3,200,000	863,000	$< 1.0 \times 10^{-5}$	1.92
Yes	History Heuristic	5,050,000	1,090,000	$< 1.0 \times 10^{-5}$	1.64
Yes	Killer Moves	3,200,000	799,000	$< 1.0 \times 10^{-5}$	1.92

findings from [17] with this knowledge in mind, such an outcome is rather predictable. We had earlier observed that, in the context of the History-ADS heuristic, the Move-to-Front rule consistently outperformed the less sensitive Transposition rule. The strategy of the History Heuristic, however, is even less sensitive to change than the Transposition rule. This is because, according to its specification, the ranking of the moves will only change when one move's counter exceeds another. As opposed to this, use of the Transposition rule leads to some change in structure every time it is queried, even if it is slight.

Our results suggest that the History Heuristic allows very strong moves to gain a substantial lead over all others. Indeed, when viewing the History Heuristic's internal updates as the search proceeded, in both the Virus Game and Othello, a single move would quickly gain a nearly insurmountable lead. This is the likely reason for the History Heuristic's poor performance, compared to the History-ADS heuristic, and is consistent with our previous observations. Our results strongly suggest that the History-ADS heuristic outperforms the History Heuristic under a broad set of board games, and we hypothesize, based on these results, that it is likely to do so in others as well.

The fact that the single ADS, Move-to-Front History-ADS heuristic outperforms the Killer Moves strategy is not at all surprising, considering our previous observations from [18]. Indeed, we had earlier determined that a single ADS would generally outperform a multi-level ADS, and that a multi-level ADS with a restriction on its length would have its performance hampered even further. As the Killer Moves technique functionally identical to the History-ADS with a multi-level ADS, and with a limit of two on the length, we would expect it to be outmatched by the single, unbounded ADS. Our results, clearly, support that.

The degree by which the Killer Moves technique was outperformed varied between the various game models, with it doing best in Chinese Checkers, and worst in Relaxed Checkers. Given that it can only maintain a very small number of moves, this suggests that storing more information achieves a superior move ordering in the case of Relaxed Checkers, but it is not so critical in the case of Chinese Checkers, with the other games falling between these extreme cases.

7 Conclusions

Our results reinforce our previous findings, that the History-ADS heuristic is able to produce strong gains in terms of tree pruning. Additionally, we have also clearly demonstrated that the History-ADS heuristic is capable of outperforming the established Killer Moves technique and History Heuristic in a wide range of game models and configurations, in some cases by a substantial margin. This is a particularly strong result, which serves to justify its usage in game playing engines, particularly given its lightweight qualities, and the fact that it does not need any additional sorting.

Our results further reinforce the idea that, in the context of the History-ADS heuristic, the most basic configuration tends to perform best. We confirm this because the single, unbound, Move-to-Front implementation of the History-ADS outperformed both established heuristics. This suggests that within the perspective of move ordering, that is based on a move history criterion, the adage "simpler is better" holds true.

References

1. A. Rimmel, O. Teytaud, C. Lee, S. Yen, M. Wang, and S. Tsai, "Current frontiers in computer go," *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, p. in press, 2010.
2. S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, pp. 161–201. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 3rd ed., 2009.
3. C. E. Shannon, "Programming a computer for playing Chess," *Philosophical Magazine*, vol. 41, pp. 256–275, 1950.

4. G. M. Baudet, "An analysis of the full alpha-beta pruning algorithm," in *Proceedings of the tenth annual ACM symposium on Theory of computing*, pp. 296–313, 1978.
5. D. E. Knuth and R. W. Moore, "An analysis of alpha-beta pruning," *Artificial Intelligence*, vol. 6, pp. 293–326, 1975.
6. J. Schaeffer, "The history heuristic and alpha-beta search enhancements in practice," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1203–1212, 1989.
7. S. Akl and M. Newborn, "The principal continuation and the killer heuristic," in *Proceedings of ACM'77 the 1977 Annual Conference*, pp. 466–473, 1977.
8. N. Sturtevant, *Multi-Player Games: Algorithms and Approaches*. PhD thesis, University of California, 2003.
9. C. Luckhardt and K. Irani, "An algorithmic solution of n-person games," in *Proceedings of the AAAI'86*, pp. 158–162, 1986.
10. M. P. D. Schadd and M. H. M. Winands, "Best Reply Search for multiplayer games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 57–66, 2011.
11. N. Sturtevant and M. Bowling, "Robust game play against unknown opponents," in *Proceedings of AAMAS'06, the 2006 International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 713–719, 2006.
12. N. Sturtevant, M. Zinkevich, and M. Bowling, "Prob-Maxn: Playing n-player games with opponent models," in *Proceedings of AAAI'06, the 2006 National Conference on Artificial Intelligence*, pp. 1057–1063, 2006.
13. G. H. Gonnet, J. I. Munro, and H. Suwanda, "Towards self-organizing linear search," in *Proceedings of FOCS'79, the 1979 Annual Symposium on Foundations of Computer Science*, pp. 169–171, 1979.
14. J. H. Hester and D. S. Hirschberg, "Self-organizing linear search," *ACM Computing Surveys*, vol. 17, pp. 285–311, 1985.
15. S. Polk and B. J. Oommen, "On applying adaptive data structures to multi-player game playing," in *Proceedings of AI'2013, the Thirty-Third SGAI Conference on Artificial Intelligence*, pp. 125–138, 2013.
16. S. Polk and B. J. Oommen, "Novel AI strategies for multi-player games at intermediate board states," in *Proceedings of IEA/AIE'2015, the Twenty-Eighth International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems*, pp. 33–42, 2015.
17. S. Polk and B. J. Oommen, "Enhancing history-based move ordering in game playing using adaptive data structures," in *Proceedings of ICCCI'2015, the 7th International Conference on Computational Collective Intelligence Technologies and Applications*, pp. 225–235, 2015.
18. S. Polk and B. J. Oommen, "Space and depth-related enhancements of the history-ads strategy in game playing," in *In Proceedings of CIG'2015, the 2015 IEEE Conference on Computational Intelligence and Games*, pp. 322–327, 2015.
19. S. Polk and B. J. Oommen, "On enhancing recent multi-player game playing strategies using a spectrum of adaptive data structures," in *In Proceedings of TAAI'2013, the 2013 Conference on Technologies and Applications of Artificial Intelligence*, 2013.
20. J. Schaeffer, N. Burch, Y. Bjornsson, A. Kishimoto, M. Muller, R. Lake, P. Lu, and S. Sutphen, "Checkers is solved," *Science*, vol. 14, pp. 1518–1522, 2007.