COMPUTERS AND OCTI: REPORT FROM THE 2001 TOURNAMENT

Charles Sutton[1]

Department of Computer Science, University of Massachusetts, Amherst, MA

ABSTRACT

Computers are strong players of many strategy games, but in some games, namely Go, success has been more elusive. Thus, it may be fruitful to explore games with intermediate complexity. Octi is such a game. I describe the game of Octi, report results from the 2001 Computer Octi Tournament, and explain why Octi may be a useful domain for research in game AI.

1. INTRODUCTION

Brute-force search has been a successful technique for many deterministic strategy games, including checkers (Schaeffer *et al.*, 1992), chess (Marsland and Schaeffer, 1990), and Othello (Buro, 1997; Lee and Mahajan, 1990). One shortcoming of the technique, acknowledged by Campbell (1996), is that play tends to be not very "human": programs can miss some strategic concepts, and they neither adapt to their opponents nor attempt to forestall their opponents from adapting to them. And scaling to a game like Go, with its tremendous branching factor, is difficult. One approach to building more human-like players is to study games whose complexity is between chess and Go, that is, to focus on games which seem to reward both brute-force search and other heuristic methods. One such game is Octi.

Octi is a strategy game for two or four players. Pieces can move, jump, and capture as in checkers, but only in directions in which a prong has been placed on a previous move. Thus, Octi challenges players to choose between attacking, defending, and developing their pieces. Because friendly pieces can jump each other, two weak pieces working together are often more valuable than one strong piece. Formations are central to Octi strategy, better for both attack and defense. Both when assembling formations and when choosing directions for prongs, decisions have long-term effects which can make brute-force search less effective. This could make Octi a good game for studying more human-like approaches to game playing.

This paper has two goals: first, to report the results of the 2001 Computer Octi Tournament, and second, to show why Octi may be a good domain for more "human-like" approaches to game AI. The paper is organized as follows. In the next section, I describe the rules of Octi. In Section 3, I report the tournament results. In Section 4, I analyze one of the tournament games in detail. In Section 5, I describe the games played by the tournament winners against Octi inventor Don Green. In the last section, I analyze why the game is an interesting problem for game AI.

2. RULES OF OCTI

Octi is like checkers on steroids. An abstract strategy game invented in 1999 by Don Green, Octi has two versions for two players: Octi-6$\times$7 and Octi-X (short for Octi-Extreme).[2] The rules of Octi-6$\times$7 are simpler, so I will describe those first, and then describe the additional rules for Octi-X.

Octi-6$\times$7 is played on a 6 by 7 board pictured in Figure 1. Each player has four starting squares, called Octi squares. The object of the game is to occupy any one of the opponent's Octi squares. Each player starts with

[1]email:casutton@cs.umass.edu

[2]Octi-6$\times$7 was originally marketed as Octi for Kids, and is now sold under the title Octi: New Edition. Octi-X was originally marketed as Octi.

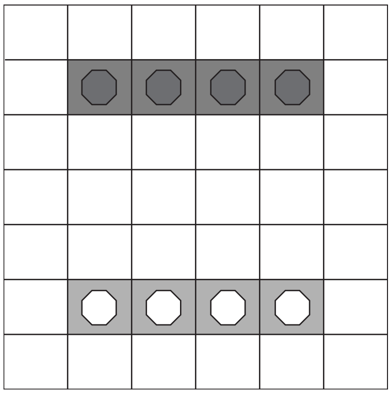

Figure 1: The starting position in Octi-6×7. The starting squares, called Octi squares, are shaded. Each player begins with 12 prongs. White wins if it occupies any of Black's starting squares, and vice versa.

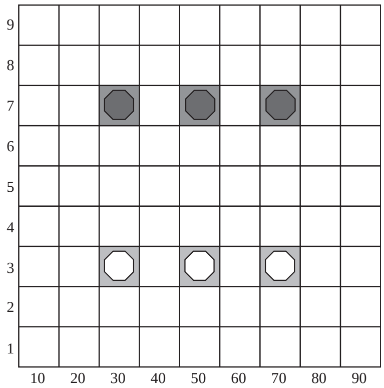

Figure 2: The starting position in Octi-X. The starting squares are shaded. Each player begins with 25 prongs and 4 reserve pods. White wins if it occupies all three of Black's starting squares, and vice versa.

four eight-sided pieces called pods and 12 small pegs called prongs. The pods have holes in each side for adding prongs. The players start with one pod on each of their Octi squares.

On each turn, a player does one of two things. First, a player can add a prong to one of their pods. Prongs cannot move once placed, and pods never rotate. Second, a player can move a pod. Pods may only move in directions in which they have prongs. A pod can either walk one square or jump over other pods as in checkers, optionally capturing them. A player can capture either friendly or enemy pods, in either case seizing their prongs. A pod cannot jump over the same square twice in one move, although it can land on the same square twice by arriving from different directions. When jumping over pieces, every square a pod lands on must be unoccupied, as in checkers.

Players cannot pass. A player who cannot move on his turn—usually because all his pieces have been captured—loses.

Octi-X is more complex. It uses a larger 9 by 9 board pictured in Figure 2. Each player has three Octi squares, the object of the game being to occupy all three. The players have more resources: they start with 25 prongs and four reserve pods.

In Octi-X, a player can add prongs and move pods as in Octi-6×7. Or, a player can add a reserve pod onto any of her Octi squares. A reserve pod can also be placed onto any enemy Octi square that the player occupies; reserve pods are good for building formations in enemy territory. If a player occupies an enemy Octi square, she can free a captured pod instead, placing it on any Octi square that she occupies.

Also, Octi-X allows pods to stack. When walking by one square, a pod may move onto a square occupied by a friendly pod; the group is called a stack. Pieces in a stack can move simultaneously, but they cannot jump over each other.

The complete rules for both games, and rules for the four-player version, are available online (Green, 2002a; Green, 2002b).

Octi strategy centers on formations, like those in Figure 3. Formations have two main advantages: they cannot be jumped as easily (because as in checkers, two pieces in a row cannot be jumped), and they can attack more squares, because the pieces can jump over each other.

Formations are important even in Octi-6×7, because if a player were to do nothing else but march one piece down the board, his opponent normally would have time to forestall the attack and launch an effective counterattack. But walking down the board is always a threat in Octi-6×7: Four pods is not enough to protect all four Octi squares at once. so some squares will be less well-defended. In Octi-X, on the other hand, having reserve pods gives the defender an advantage around his Octi squares, so a single pod cannot usually make a strong assault.

Octi presents a complex search space. Based on a sample of games played online, the average branching factor of Octi-6×7 is about 31 (min 2, max 101), and Octi-X 39 (min 2, max 626).[3] For comparison, a commonly-cited branching factor for checkers is 3, Othello 10, and chess 36. But in Octi more than other games, players can act to increase the branching factor. Two situations lead to combinatorial explosion in the number of legal moves: jumping over pieces, because for each square a piece jumps over, the player can choose whether to capture; and stacks, because each of the pieces in a stack can move simultaneously. So by using formations and stacks, players can ratchet up the branching factor in a game, perhaps as part of an anti-computer strategy.[4] In fact, because of stacking, there is an Octi-X position with $7^8 - 1$ legal moves, all of which lead to distinct board positions! Fortunately, this position is extremely unlikely to arise in practice.

3. THE TOURNAMENT

The second Computer Octi Tournament was held in New Haven, CT, on August 25, 2001. The tournament had two sections: one for Octi-X and one for Octi-6×7. The tournament was the second annual for Octi-X, and the first computer tournament for Octi-6×7. Four Octi-6×7 programs were entered (Table 1) and two Octi-X (Table 2). All entrants used some variant of alpha-beta search. Some features used in evaluation functions included distance to the opponent's Octi squares, mobility, number of adjacent pieces, and material advantage.

Both sections were round-robin. All the programs were run on identical Windows PCs in the Yale University Statistics Lab, except for Haskellbot, which ran on a Macintosh PowerBook G3. Each program had two hours for the game. For Octi-6×7, this left about six minutes per move, because Octi-6×7 games rarely last more than 20 moves. Octi-X games can take much longer, but rarely more than 100 moves.

In Octi-6×7, AaronBot swept the field, winning all six games. Both HAL and Casbah were competitive, but Casbah was out-searched by AaronBot in several key positions, and HAL ignored the time controls, losing most of its games on time. It is difficult to compare the programs, however, because some games were cancelled when it became clear that AaronBot had won. The full results are given in Table 3.

In Octi-X, the entrants were Casbah, by Charles Sutton, and ChiouBot, by Charles Chiou. This was a rematch of the previous year's tournament, where Casbah had beaten ChiouBot. However, this year ChiouBot was rewritten and greatly improved, while Casbah had apparently been broken when Sutton developed the Octi-6×7 version. ChiouBot won handily, even occupying all three bases (previous Octi-X programs have won by capturing all the opponent's pieces). I will analyze a position faced by ChiouBot in Section 5.

4. A SAMPLE GAME

The most interesting game of the tournament was the Octi-6×7 game between HAL and AaronBot. HAL lost this game on time, but it was unofficially played out. The game was unique because the players ran out of prongs, which makes the final few prongs and *zugzwang*[5] important. In this section, I analyze the game in detail.

[3] I thank Aaron Armading for providing this data.
[4] I am indebted to Don Green for suggesting this point.
[5] A position is *zugzwang* when the player to move would be fine if he could pass, but all legal moves worsen the position.

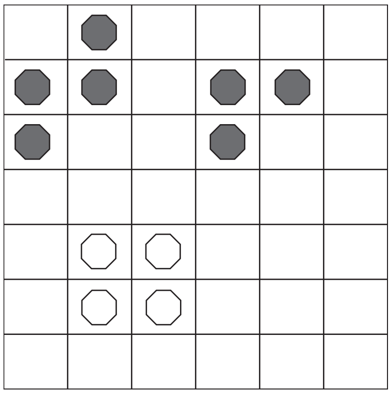

Figure 3: Three common Octi formations, shown without prongs.

Program	**Author**	**Programming Language**
AaronBot	Aaron Armading	C++
Haskellbot	Thomas Hallock	Haskell
Casbah	Charles Sutton	Common Lisp
HAL	Charles Tabony	C

Table 1: Tournament entrants in Octi-6×7.

Program	**Author**	**Programming Language**
ChiouBot	Charles Chiou	C++
Casbah	Charles Sutton	Common Lisp

Table 2: Tournament entrants in Octi-X.

	AaronBot	Haskellbot	Casbah	HAL	**TOTAL**
AaronBot		2–0	2–0	2–0	6–0
Haskellbot	0–2		0–1	0–1	0–4
Casbah	0–2	1–0		2–0	3–2
HAL	0–2	1–0	0–2		1–4

Table 3: Results from the Octi-6×7 tournament. Some games that could not affect the outcome of the tournament were not played.

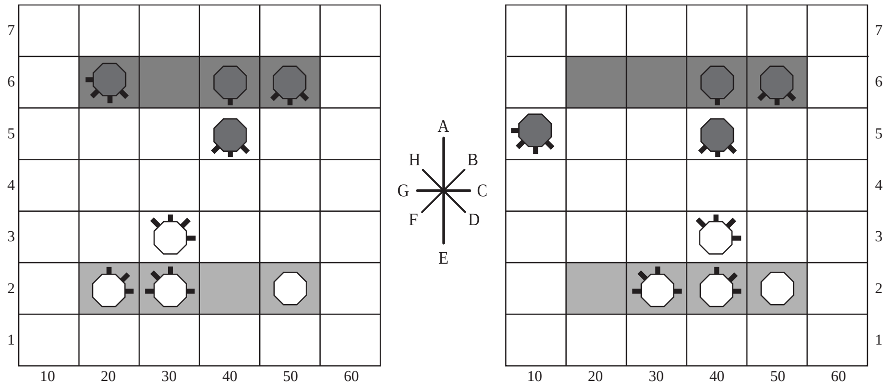

Figure 4: HAL (White) versus AaronBot (Black). At left, move 13, white to move. Neither player has been able to attack the other's weakness, and now they are running out of resources. Each has only one prong left. At right, move 14, black to move. White has restructured his formation, but Black has a potent flank attack.

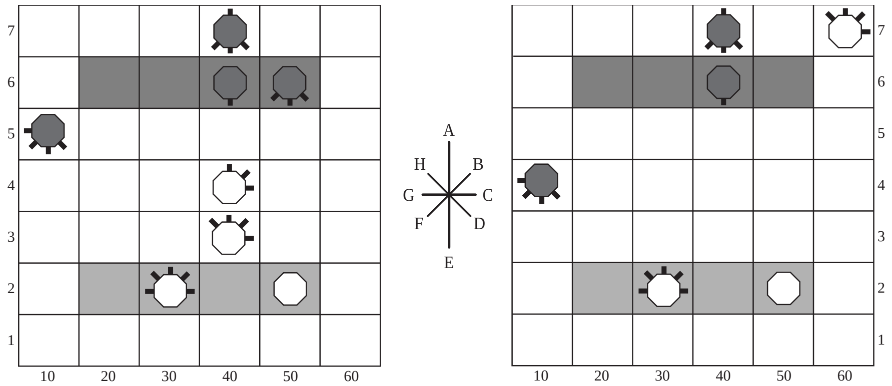

Figure 5: HAL versus AaronBot. At left, move 16, Black to move. At right, move 17, Black to move. Now Black is lost. Notice that white captured its own pod on 44 to prevent Black from countering 47-45-43x-21x.

Early in the game, both programs made opposing L-formations.[6] The L-formation is reasonably strong, but it has a major weakness: the inside of the L—here, squares 23 and 55. If an enemy pod could jump to square 23, for example, it could capture the whole formation in one jump, for example, with a jump that ends 23-43x-21x-23x (assuming, of course, that the enemy had the appropriate prongs). In this position, however, the weakness is inaccessible: White's formation is weak on the left, but all of Black's power is on the right, and vice versa. So for the first twelve moves both sides have been adding prongs without making many real threats. Some of these prongs have probably been inadvisable, such as the g prong on 26. This pod is almost on the left edge already, so moving further left is not helpful. Also, if needed pod 26 can move to the left edge using its f prong.

But by move 13 (see Figure 4), the programs are down to one prong each. Once they run out, the only available moves will be jumps and walks, until the players regain prongs by capturing a piece. If White adds its last prong, it must take care to avoid *zugzwang* on the next couple moves—it is not good to be the first to run out of prongs. But White does something interesting, shifting its formation with 13. 22-42 and 14. 33-43 (see Figure 4). Meanwhile Black has used the extra time to threaten to run down the side with 13. . . . 26-15.

White has moved its formation one space to the right. The empty pod on square 52 is important because it

[6] Octi notation is described online (Green, 2002a; Green, 2002b). The board is numbered from 11 to 67 and the directions from 'a' to 'h' as shown in the board diagrams. An add prong move is written like 22+a. A jump move is written like 55-37-15. An 'x' indicates a capture, for example, the move 43-45x-67x means that the pod on square 43 moved to square 67, capturing the pieces on squares 44 and 56.

The full game of HAL v. AaronBot was **1.** 22+a 36+d **2.** 22+c 46+e **3.** 42+h 36-45 **4.** 42-33 56+e **5.** 33+b 45+e **6.** 33+a 56+f **7.** 32+a 45+f **8.** 32+c 56+d **9.** 22+b 26+f **10.** 32+g 26+g **11.** 33+c 26+d **12.** 32+h 26+e **13.** 22-42 26-15 **14.** 33-43 45+a **15.** 32+b 45-47 **16.** 42-44 15-14 **17.** 43-45x-67x 14-13 **18.** 32-22 47-56 **19.** 67+f 46-45 **20.** 67+g. At this point, Black made an illegal move, but the position was lost anyway.

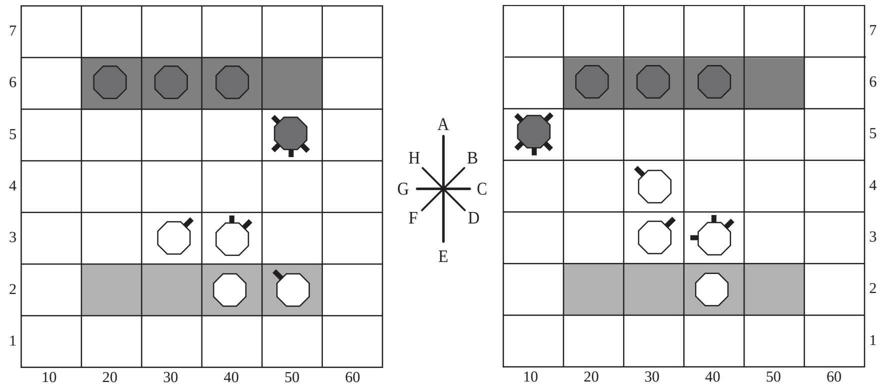

Figure 6: Game between AaronBot (White) and Octi creator Don Green (Black). At left, move 6, with Black to move. At right, move 8, Black to move. Black is lost.

prevents a potential game-winning jump 56-34-52; empty pods often fill such roles. This four-piece formation is strong, impregnable to Black's available forces. It is Black's move, and Black must address White's potential attack, 32+b. Black defuses this with 14. . . . 45+a, setting up an avenue for retreat. White attacks anyway, 15. 32+b, and Black retreats 15. . . . 45-47. Now White is out of prongs, and must move a pod. White charges up the center with 16. 42-44 (see Figure 5).

Black cannot do much here. If Black tries to advance into the center, he would lose immediately. Advancing down the right is too slow. And Black is out of prongs, and so must move a pod. Black continues its run down the left side with 16. . . . 15-14, perhaps the best that it can do. White charges toward the Black Octi squares with 17. 43-45x-47x (Figure 5). Note that White captures its own piece on 44. This is necessary because otherwise Black would use square 44 as a springboard to make a game-winning jump 47-45-43x-21x. Usually when a player captures his own pod, it is to prevent a devastating counter-jump like this.

Now Black can move toward his opponent's Octi squares with 14-13, but White would block 32-22 and Black would need three moves to win: one to capture, one to add a back prong (gained after capturing the sacrificial White pod at 22), and one to move onto an Octi square. But White needs only two moves to win. The useless prong 14g is now painful: If that prong pointed backward, Black would have prevailed. As it was, White won three moves later. A bug in AaronBot (Black) caused it to think it had prong when it did not, so it made an illegal add-prong move. But as we have seen, the position was lost anyway.

Running out of prongs, it should be noted, is not as crucial in Octi-X, because there are more prongs and more pods. More pods tend to capture each other more often, and players seize prongs when they capture. However, having more prongs than one's opponent is still an advantage.

In the positions I have analyzed, Black seems to have done about the best possible, suggesting that the original position in move 13 (Figure 4) was already a loss for Black. This has since been proven by computer analysis, but it was not apparent to either the computers or the human observers during the tournament.

It is important to keep in mind that White had a big advantage in the unofficial game: White ignored the clock. In the official record, White lost on time. But this is still a good example of play in Octi-6$\times$7.

5. PLAYING AGAINST OCTI'S INVENTOR

Both tournament winners played Don Green, Octi's inventor and one of its best players. To make the games more sporting, Green allowed the programs about a day per move. The Octi-6$\times$7 program AaronBot won, while the Octi-X program ChiouBot lost.

Figure 6 shows a key sequence in the AaronBot versus Green game. Don Green has set up an interesting offense that threatens to flank on both sides. Of course Green can attack on the right, but also he can still attack on the left by leaping the board with 55-37-15. White has opted for a more solid four-piece formation which is slower, but has good attacking potential. Perhaps underestimating White's attack, Green develops

6. . . . 55+b. But AaronBot can attack up the middle, and it does a move later with 8. 52-34. Now (Figure 6), Green has lost. He needs at least three moves to win by running down the side, while White has a threat to win in two. Green delayed the loss, but succumbed a few moves later. This is a prime example of four weak pieces working together being better than one strong piece. Stung by losing in a favorite offense, after the game Green found a winning variation by setting up some defense with c, d, and f prongs on square 36.

Figure 7 shows the game of ChiouBot versus Green, with ChiouBot to move. On the previous move, Green had just moved three pieces out of a stack on square 33 to squares 34, 44, and 43, forming an instant square formation (recall that in Octi-X all pieces in a stack can move simultaneously). Black can capture the piece on 53, but this is a trap: taking the piece will leave Black split up and defenseless. This is a common tactic in Octi-X, offering a piece to break up the opponent's position. Black takes the bait, moving 10. . . . 46df-64x-42x, even capturing its own piece. The self-capture is not a good idea, either; possibly it was to prevent White jumping 44-88, but this is not White's best attack. The game is essentially over now.

6. SUMMARY

Play was much stronger in Octi-6×7 than Octi-X. In part this may be because more people worked on programs for the small board. But more likely, Octi-6×7 is just a simpler game—it was designed that way, as a children's version of Octi-X. First, Octi-6×7 is more tactical, because the Octi squares are closer together, and because the game is played for only one base. Also, formations can be built more easily in Octi-6×7: the pieces start closer together, so, for example, you can build a three-piece L formation in two moves. This means that Octi-6×7 programs can play using formations without much special support in the evaluation function, such as pattern-matching to identify formations. To build a strong formation in Octi-X, on the other hand, a player has to add and develop several new pods. Current Octi-X programs are not good at setting up formations or exploiting weaknesses in the opponent's formations.

The difficulties in programming Octi-X are twofold: where to put pods (i.e., what formations to use) and where to put prongs. Both of these are—for humans, at least—strategic decisions with long-range consequences. In many games some strategic decisions can be handled by opening and end game databases. These are less useful in Octi-X, for the opening is very much like the middle game and end game: in all phases of the game, the strategy is to attack and defend in formations, and setting up a formation in the opening is much like setting up a formation to protect your last home square.

Although the average branching factor of Octi-X is only slightly higher than chess, in Octi moves have long-range consequences which make full-width search less effective. Moves in other games have long-range consequences too, such as pawn structure in chess and control of the edges in Othello. It could be that strong Octi-X programs will use essentially the same heuristic search techniques that have succeeded in chess, Othello, and Octi-6×7. Octi-X's branching factor is not so high that this cannot happen, although special techniques may be needed for combinatorially explosive situations such as stacking and choosing which pieces to capture.

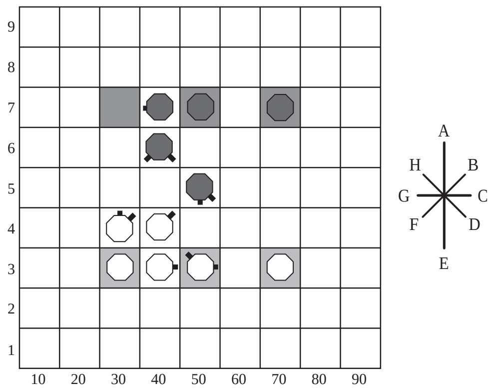

Figure 7: Octi-X game between Don Green (White) and ChiouBot (Black). Move 10, ChiouBot to move. ChiouBot is about to move 46df-64x-42x, which captures a piece at the expense of ruining its position.

So Octi is far from solved. But very good programs exist for Octi-6$\times$7, one of which has beaten the game's designer, although not under tournament conditions. Programs for Octi-X—including the author's—are not so good. In Octi, search is very important, but some important decisions in the game may be more amenable to strategies such as pattern-matching and planning. Thus Octi might serve as a bridge between chess and Go, that is, a game in which both brute-force and more human-like methods are needed to win.

ACKNOWLEDGMENTS

I thank Don Green and Jodi Schneider for providing helpful comments on drafts of this paper. Also, I thank the tournament participants—Aaron Armading, Thomas Hallock, Charles Tabony, and Charles Chiou—for answering questions about their programs, and for being worthy (and friendly) adversaries.

7. REFERENCES

Buro, M. (1997). The Othello match of the year: Takeshi Murakami vs. Logistello. *Journal of the International Computer Chess Association*, Vol. 20, No. 3, pp. 189–193.

Campbell, M. S. (1996). "An Enjoyable Game": How HAL Plays Chess. *HAL's Legacy: 2001's Computer as Dream and Reality* (ed. D. G. Stork). MIT Press.

Green, D. (2002a). Rules to Octi-Extreme. http://www.octi.net/.

Green, D. (2002b). Rules to Octi: New Edition. http://www.octi.net/.

Lee, K.-F. and Mahajan, S. (1990). The Development of a World Class Othello Program. *Artificial Intelligence*, Vol. 43, No. 1, pp. 21–36.

Marsland, T. and Schaeffer, J. (eds.) (1990). *Computers, Chess, and Cognition.* Springer Verlag.

Schaeffer, J., Culberson, J., Treloar, N., Knight, B., Lu, P., and Szafron, D. (1992). A world championship caliber checkers program. *Artificial Intelligence*, Vol. 53, Nos. 2–3, pp. 273–290.