EVOLVING COMPLEX OTHELLO STRATEGIES USING MARKER-BASED GENETIC ENCODING OF NEURAL NETWORKS

David Moriarty and Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin, Austin, TX 78712-1188
moriarty,risto@cs.utexas.edu

Technical Report AI93-206
September 1993

Abstract

A system based on artificial evolution of neural networks for developing new game playing strategies is presented. The system uses marker-based genes to encode nodes in a neural network. The game-playing networks were forced to evolve sophisticated strategies in Othello to compete first with a random mover and then with an α-β search program. Without any direction, the networks discovered first the standard positional strategy, and subsequently the mobility strategy, an advanced strategy rarely seen outside of tournaments. The latter discovery demonstrates how evolution can develop novel solutions by turning an initial disadvantage into an advantage in a changed environment.

1 Introduction

Game playing is one of the oldest and most extensively studied areas of artificial intelligence. Game playing appears to require sophisticated intelligence in a well-defined problem where success is easily measured. Games have therefore proven to be important domains for studying problem solving techniques.

Most research in game playing has centered on creating deeper searches through the possible game scenarios. Deeper searches provide more information from which to evaluate the current board position. This approach, however, is different from the play of human experts. In a psychological study, DeGroot (1965) found that game playing experts did not search a larger number of alternative moves than novices. Experts in fact use a much greater knowledge of the game together with sophisticated pattern recognition to focus the search (Charness 1976; Frey and Adesman 1976).

This paper presents a more "human-like" approach to game playing using artificial life techniques. Artificial neural networks and genetic algorithms were combined to evolve artificial game-playing "creatures". The creatures were required to learn the game of Othello without any previous knowledge of the game. Without hand-coded rules or heuristics, the creatures were free from any bias in their decision of where to move. The strategies evolved purely from discovery. Creatures would learn the rules and how to play well simply through playing the game.

The creatures were afforded no search mechanism, and were thus forced to use pattern recognition pertaining only to the current board pattern to achieve good play. The goal was to see what strategies the creatures would develop. The creatures were first evolved against a random mover, and quickly developed a positional strategy usually found in beginners. Another strategy, mobility strategy, is known to exist that is very hard to learn but produces much stronger play. After a positional strategy was encoded into an α-β search program and the creatures were allowed to compete with the α-β, they evolved to exploit their initial material disadvantage and discovered the mobility strategy.

The creature's neural networks were encoded genetically based on a marker-based approach originally proposed by Fullmer and Miikkulainen (1992). For an empirical comparison, another population of creatures was evolved using a fixed architecture encoding scheme; however, only the marker-based scheme turned out sufficiently powerful in this task.

The first section reviews the game of Othello. The rules are presented for the reader not familiar with the game. The next section discusses the application of the genetic algorithm to this problem and the encoding of the neural networks in the marker-based and fixed architecture schemes. Section 4 presents the main experimental results. The significance of evolving the mobility strategy is discussed in section 5 and further research is proposed.

2 The Game of Othello

2.1 Previous work

Othello has traditionally been very popular in Japan, second only to Go. It was introduced to America in the early 1980's and soon attained international popularity. It is enjoyed by novices and experts, for its rules are simple, but complex strategies must be mastered to play the game well.

Rosenbloom created one of the first Othello programs called Iago (Rosenbloom 1982). This program, which achieved master level play, was based on α-β search techniques with kill tables. Lee and Mahajan developed a successor to Iago named Bill (Lee and Mahajan 1990). Bill was based on similar search techniques, but also implemented Bayesian learning and many look up tables to recognize common patterns. While Bill does use more knowledge in evaluating board positions, its backbone is still the

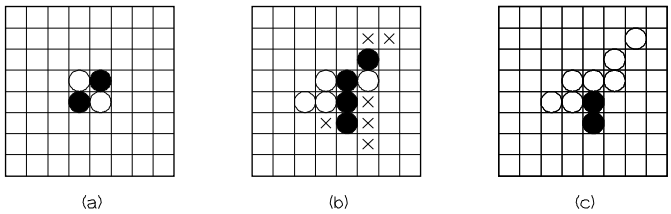

Figure 1: The Othello board: (a) The initial setup. (b) After four moves. The legal moves for white are marked with X's. (c) After white has moved to the rightmost square.

α-β search.

2.2 Rules of Othello

Othello is a two-player game played on an 8x8 board. All pieces (or tiles) are identical with one white side and one black side. The initial board setup is shown in Figure 1(a). Each player takes turns placing pieces on the board with his color face up. A player may only move to an open space that causes an opponent's piece or pieces to be surrounded by the new piece and another one of the player's own pieces. Pieces may be captured vertically, horizontally, or diagonally. Figure 1(b) shows the legal moves for white for the given board pattern. Once a move is made, the captured pieces are flipped. Figure 1(c) shows the board layout resulting from a move by white in the second row of the seventh column. The game is continued until there are no legal moves available for either player. If a player has no legal move, he must pass. The winner is the player with the most pieces in the final board configuration.

2.3 Strategies

A game of Othello can be broken down into three phases: the beginning game, the middle game, and the end game. The beginning game can be adequately handled by an opening book. The end game is simply played by maximizing your pieces while minimizing your opponent's. The strategy for the middle game, however, is much more elusive. The goal of the middle game is to strategically position your pieces on the board so that (a) they cannot be captured by your opponent and (b) they provide a good vehicle to capture your opponent's pieces during the end game.

There are two basic middle-game strategies in Othello. The positional strategy stresses the importance of specific positions on the board. Spaces such as corners and edges are considered valuable, while others are avoided. Corners are especially valuable because once taken, they can never be recaptured. Normally, a person using a positional strategy tries to maximize his valuable pieces while minimizing his opponent's. This is the typical strategy for beginners since it is easily understood and implemented.

The mobility strategy introduces the number of available moves for each player into the evaluation. The goal is to maximize the number of moves you have to choose from while minimizing your opponent's choices. This strategy stresses keeping the number of your tiles low during the middle game. By reducing the number of your pieces you are also reducing the number of legal moves for your opponent. Proper use of the mobility strategy can force an opponent to make a bad move because it is the only available move. All tournament players employ this strategy to some degree.

The mobility strategy has been shown to be much harder to learn than the positional strategy (Billman and Shaman 1990). Unlike many good ideas that are often discovered independently by several people, it is widely believed that the mobility strategy was discovered only once in Japan and has since been introduced to America and Europe through American players in contact with Japanese groups (Billman and Shaman 1990). Being able to independently discover the mobility strategy through artificial evolution would therefore be a significant demonstration of the potential power of the evolutionary approach.

3 Implementation

3.1 Game-playing Neural Networks

Our approach was to evolve a population of neural networks in the game of Othello. Each network sees the current board configuration as its input and indicates the goodness of each possible move as the output. In other words, instead of searching through the possible game scenarios for the best move, the neural networks rely on pattern recognition in deciding which move appears the most promising in the current situation.

For each board space there are three possible states: it may be occupied by the creature's piece, it may be occupied by the opponent's piece, or it may be unoccupied. For each board space, two input units were used. If the first unit is on, the space is occupied by the player's piece. If the second input unit is on, the space is occupied by the opponent's piece. If they are both off, the space is unoccupied. The two input units are never both on. The total number of input units was therefore 128.

Each network contained 64 output units. Each output unit corresponded directly with a space on the board. The activity of each output unit was taken to indicate how strongly the network suggested moving into that space.

The network architectures were encoded in artificial chromosomes and evolved through genetic algorithms (Goldberg 1988; Holland 1975). Each chromosome was a string of 8-bit integers ranging from -128 to 127. Two different encoding schemes were implemented: marker-based and fixed architecture encoding.

3.2 Marker-based Encoding

The marker-based scheme is inspired by the biological structure of DNA. In DNA, strings of nucleotide triplets specify strings of amino acids that make up a protein. Since multiple proteins may be defined on the same DNA strand, certain nucleotide triplets have a special status as

$< start >< label >< value >< key_0 >< label_0 >< w_0 > ... < key_n >< label_n >< w_n >< end >$

$< start >$	- Start marker.
$< label >$	- Label of the node.
$< value >$	- Initial value of the node.
$< key_i >$	- Key that specifies whether connection is from/to an input/output unit or from another hidden unit.
$< label_i >$	- Label of the unit where connection is to be made.
$< w_i >$	- Weight of connection.
$< end >$	- End marker.

Figure 2: The definition of a node in marker-based encoding

markers that indicate the start and the end of a protein definition (Griffiths et al. 1993).

Artificial genes can similarly use markers to define separate nodes in a neural network. Each node definition contains a start integer and an end integer. The integers in-between specify the node. Figure 2 shows the node definition in the marker-based scheme. The start and end markers are determined by their absolute value. If the value of an integer MOD 25 equals 1, it is a start marker. If the value MOD 25 equals 2, it is an end marker. Any integer between a start marker and an end marker is always part of the genetic code. Therefore, an integer whose value MOD 25 equals 1 may exist between two markers and will not be treated as a marker.

Since 8-bit integers for the connection labels only allow for 256 distinct connection labels, using 128 for the input units and 64 for the output units would only leave 64 labels for the hidden units. To avoid this restriction, in our implementation of the marker-based system each connection label consists of two 8-bit integers. The *key* integer specifies whether the connection is to be made with the input/output layers or with another hidden unit. If the key is positive, the second integer, *label*, specifies a connection from the input layer (if the label is > 0) or to the output layer (if the label is < 0). If the key is negative, the label specifies an input connection from another hidden unit.

The chromosome is treated as a continuous circular entity. A node definition may begin on one end of the chromosome and end on the other. The final node definition is terminated, however, if the first start marker is encountered in the node definition. Figure 3 shows an example gene and the network information it encodes.

The nodes are evaluated in the order specified on the chromosome. Hidden units are allowed to retain their values after each activation. This allows the networks to possibly use their hidden nodes as short-term memory.

The power of the marker-based scheme comes from its ability to evolve the network architecture along with the weights. Most neural network encoding schemes fix the architectures to be evolved (Belew et al. 1991; Jefferson et al. 1991; Werner and Dyer 1991; Whitley et al. 1990). Each position on the chromosome corresponds directly to a weight in the network. The marker-based system, however, evaluates alleles according to their position relative to a start marker, not their absolute position on the chromosome. This allows for the evolution of genetic schemas that provide the maximum benefit for the creatures. Some environments may be best served by a large number of hidden units with few connections per unit. Other environments may require less hidden units with a larger number of connections per unit. The marker-based encoding can adapt the network architecture to best fit the environment.

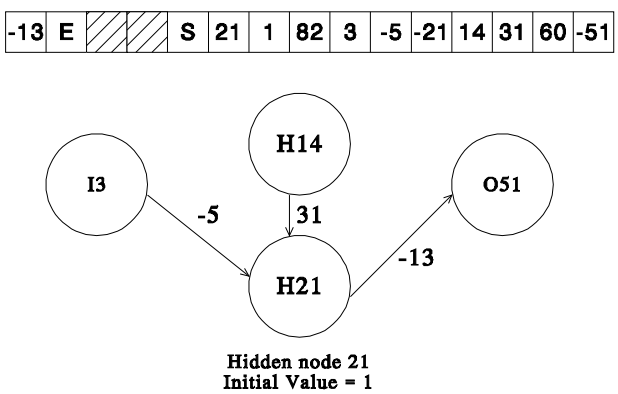

Figure 3: An example node definition in a marker-based gene. For example, the first connection has $key = 82$, $label = 3$, $w = -5$. The key and label are both positive so the connection is to be made from input unit 3.

The marker-based scheme presented above is a successor of Fullmer and Miikkulainen (1992). Whereas they defined output nodes explicitly like any other nodes in the network, our new version of marker-based encoding only requires the definition of hidden nodes. The connections to output units are given as part of the hidden node definitions, resulting in a more compact encoding when the output layer is large.

3.3 Fixed Architecture Encoding

For an empirical comparison of encoding schemes another population of game-playing creatures was evolved with a fixed architecture encoding. The networks consisted of three layers (input, output, and one hidden layer) which

were fully connected. To keep the chromosomes the same length as with the marker-based encoding, 40 hidden units were used. Each hidden unit had a recursive connection to itself to allow short-term memory to develop. The chromosome was simply the concatenation of the weights in the network. Similar fixed architecture encoding techniques have been shown to be effective in domains such as evolving communication (Werner and Dyer 1991), processing sonar sinals (Montana and Davis 1988), and trail-following (Jefferson et al. 1991).

3.4 Evolution

Populations of 50 creatures were evolved with a gene size of 5000 integers for the marker-based creatures and 5160 integers for the fixed architecture creatures. A two point crossover was employed to generate two offspring per mating. Mutation, at the rate of 0.4%, occurred at the integer level rather than the bit level. A random value was added to an allele if mutation was to occur.

The networks were given the current board configuration as input. Out of all output units that represented a legal move in a given situation, the one with the highest value was selected as the creature's move. In other words, the creatures were not required to decide which moves were legal, but only to differentiate between legal moves. This strategy turned out to speed up the evolution a great deal, while still allowing the creatures to evolve good game-playing skills.

The networks were initially evolved against a random move maker and later against an α-β search program. The number of wins over six games played determined the fitness of each network. The α-β program was allowed to search three levels down and used a positional strategy similar to Iago's (Rosenbloom 1982). Iago's heuristic also contained a complex mobility strategy, which was purposely left out to provide a weakness that the creatures could exploit.

4 Results

Experiments on artificial evolution are computationally very expensive, and many different populations were evolved throughout this research. Typically, populations required one week of CPU time on a Sun Sparc 1 workstation to evolve significant behavior.

The marker-based creatures playing against a random mover immediately began to develop a positional strategy. Whenever possible they would maximize their edge pieces, while trying to minimize those of their opponent's. The creatures' middle games consisted of taking edge and corner pieces with little regard to the middle squares. Besides recognizing valuable spaces, the creatures also learned which spaces should be avoided like those next to corners. These positions are undesirable if the corner is not already captured because they can directly lead to corner capture by your opponent. Within 100 generations, the creatures were able to beat any random mover.

The fixed architecture creatures had considerably more trouble developing a good strategy. They finally did begin to develop a positional strategy, but it took considerably more evolution time (around 1000 generations).

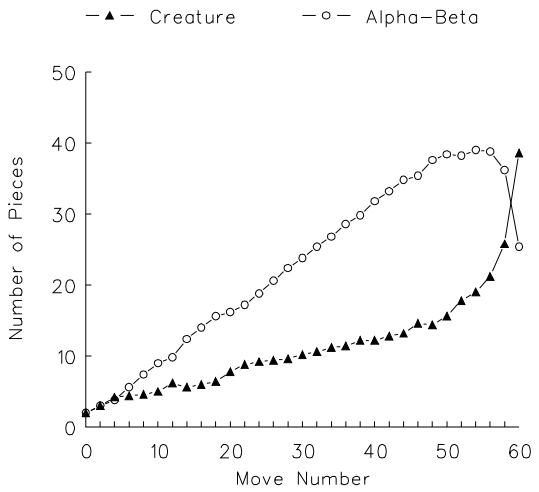

Figure 4: The average number of pieces through the game.

The creatures were then evolved against the α-β program. Initially they performed very poorly. As the populations evolved the best marker-based creatures' performance began to improve. After 2000 generations the best marker-based creatures were winning 70% of their games against the α-β. The best fixed architecture creatures could only defeat the α-β 7% of the time.

An analysis of the marker-based creatures' games showed that the creatures' strategies had changed drastically. They were not defending the edge pieces as ruthlessly as before. Instead, their attention and moves focused on the middle squares. However, their defense of corner pieces had not changed, and if presented a legal corner move the creatures would immediately capture it.

To better determine how the creatures were playing, ten games in which a creature won were chosen at random and analyzed. Figure 4 shows the average number of pieces at each point of the game for the creature and α-β during the ten games. The results show that the creatures are playing a remarkably different middle game than the α-β's. While the α-β is taking a greedy positional approach maximizing its positions, the creatures are keeping the number of their pieces relatively small. The turning point in the games come around move 54. This is a typical transition to the end game. As shown in figure 4 the creature's pieces dramatically increase while the α-β's decrease.

Such a dramatic change could only have resulted from the α-β making a vulnerable move. Since the α-β does not normally decrease its positions, it must have had no alternative. Figure 5 shows that during the crucial middle and end games the creature had an average of twice as many moves to choose from than the α-β.

Figure 6 shows an actual game the authors played against one of the creatures. The creature was playing white. Figure 6(a) shows the board configuration near the

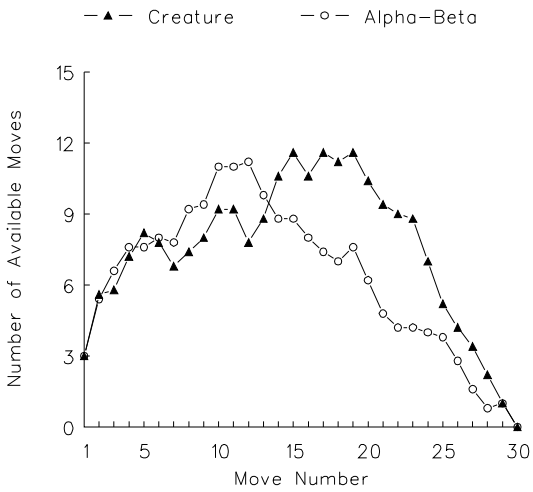

Figure 5: The average number of available moves for each player.

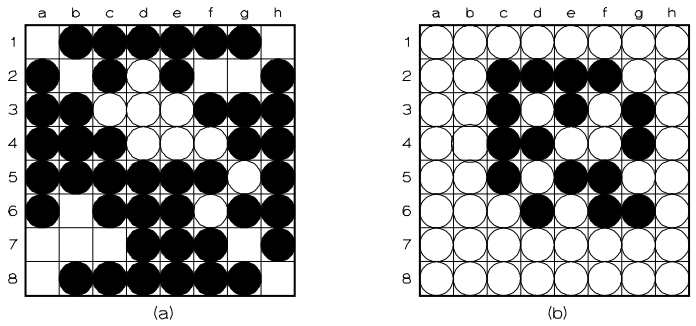

Figure 6: (a) A situation in the middle game against a creature playing white. (b) The final board configuration.

end of the middle game with white to move next. From a positional sense, the game looks very good for black. Black has conquered almost all of the edge pieces (although no corners) and is dominating the board. However while black's positions are strong, his mobility is very limited. After white moves to f2 the only legal moves for black are b2, g2, and g7. Each one of these moves will lead to a corner capture by the creature. By limiting the available moves (mobility), the creature forces black to make a bad move. Figure 6(b) shows the final board. The creature has taken every corner and captured all of black's edge pieces. The final score was 48-16 in the creature's favor.

The statistical results and the analysis of individual games clearly indicate that the creature is using the mobility strategy to defeat the weaker positional strategy.

5 Discussion

One interesting conclusion from these experiments is that the fixed architecture encoding was not powerful enough to generate sophisticated game-playing behavior. The marker-based creatures were much more adaptive to new changes in the environment. They were able to change their positional strategy to a mobility strategy after their opponents got better. The fixed architecture creatures could not make such adjustments.

A possible explanation is in the resulting architectures. In the fixed encoding, the architecture was limited to full connectivity with 40 hidden units. The marker-based creatures, however, could adjust their architecture to represent knowledge. For example, some hidden units had no inputs and effectively served as bias units. The average number of hidden units in the marker-based creatures was 115. Since some of the alleles in the marker-based chromosome serve as markers or labels, the fixed architectures actually had more connections. Since the marker-based creatures were not biased towards any architecture, it would seem that playing Othello (and possibly all game playing) is better served by a large sparsely connected hidden layer.

Since the creatures were not required to generate legal moves, at first glance it seems that they would not evolve to play the game correctly. For example, often a creature would output a high value (e.g. in a corner), which would be a good move, but unfortunately illegal in the current configuration. Interestingly, such behavior is very similar to human play. Humans will often look at a board position and think "I would love to be able to move there, but it's illegal." Humans realize that certain moves, such as corners in Othello, are almost always advantageous and keep them in mind until they become legal. It seems that the creatures' play goes through a similar process.

The creatures develop a positional strategy very much like human beginners. Most novices recognize the importance of corner pieces early on and somewhat later that of other pieces that can also be stabilized. Such positional strategy is very concrete and easy to understand. It is also easier to learn and sufficient to defeat a random mover and was thus employed by the creatures early on.

The mobility strategy is difficult for humans to discover because it is counterintuitive. Intuitively, most human game players feel they should try to conquer as much territory as possible. The artificial creatures have no such bias, and are much more flexible in considering alternative strategies.

The mobility strategy evolved quite naturally as a response to the environment. The α-β search program used a strong positional strategy and was allowed to search several moves ahead. The creatures' positional strategy was developed against a random mover and was not nearly as sophisticated. As a result, the creatures' piece count was quite low throughout the game. However, the creatures discovered that they could often win in such situations by improving their mobility instead of their positional game. Effectively the creatures used the α-β's strategy directly against the α-β. In other words, evolution was able to turn the initial disadvantage into a novel advantage. A similar process often appears to take place in successful natural evolution and adaptation into a changing environment.

Such adaptation of strategy is quite different from human beginners. Human novices often try to mimic the strategies of better players to achieve better play. A hu-

man in the same situation would have tried to improve his positional strategy to make the games closer. As a result, novel strategies are not easily discovered. If artificial evolution had been available in the early 1970's, the positional strategy could have been shown vulnerable years before any human discovered its weaknesses.

Discovering a known counterintuitive strategy demonstrates the power of the evolutionary approach. It could be extended to games such as Chess, Go, and Backgammon, where the best humans can still defeat the best computers. If novel strategies are identified, they can be used to build better heuristics. This approach is certainly not limited to game playing. Domains like natural language processing, planning, and automatic theorem proving also rely on a great deal of search. By forcing artificial creatures to compete with current heuristic search methods, better evaluation techniques could be evolved.

6 Conclusion

Marker-based encoding of neural networks proved to be more effective in adapting to changes in the environment than fixed architecture encoding. The experiments show that the marker-based system can find new problem solving techniques by evolving neural networks against existing methods. By adapting to a changing environment, an Othello strategy that eluded expert players for years was evolved in a week. It should be possible to use this same approach to find new strategies and heuristics in other domains as well.

References

Belew, R. K., McInerney, J., and Schraudolph, N. N. (1991). Evolving networks: Using genetic algorithm with connectionist learning. Technical Report CS90-174, The University of California at San Diego, La Jolla, California 92093.

Billman, D., and Shaman, D. (1990). Strategy knowledge and strategy change in skilled performance: A study of the game othello. *American Journal of Psychology*, 103:145–166.

Charness, N. (1976). Memory for chess positions; resistance to interference. *Journal of Experimental Psychology*, 2:641–653.

DeGroot, A. D. (1965). *Thought and Choice in Chess*. The Hague, The Netherlands: Mouton.

Frey, P. W., and Adesman, P. (1976). Recall memory for visually presented chess positions. *Memory and Cognition*, 4:541–547.

Fullmer, B., and Miikkulainen, R. (1992). Evolving finite state behavior using marker-based genetic encoding of neural networks. In *Proceedings of the First European Conference on Artificial Life*. Cambridge, MA: MIT Press.

Goldberg, D. E. (1988). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.

Griffiths, A. J. F., Miller, J. H., Suzuki, D. T., Lewontin, R. C., and Gelbart, W. M. (1993). *An introduction to Genetic Analysis*. W. H. Freeman.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Ann Arbor, MI: University of Michigan Press.

Jefferson, D., Collins, R., Cooper, C., Dyer, M., Flowers, M., Korf, R., Taylor, C., and Wang, A. (1991). Evolution as a theme in artificial life: The genesys/tracker system. In Farmer, J. D., Langton, C., Rasmussen, S., and Taylor, C., editors, *Artificial Life II*. Reading, MA: Addison-Wesley.

Lee, K.-F., and Mahajan, S. (1990). The development of a world class othello program. *Artificial Intelligence*, 43:21–36.

Montana, D. J., and Davis, L. (1988). Training feedforward neural networks using genetic algorithms. Technical report, BBN Systems and Technologies, Inc., Cambridge, MA.

Rosenbloom, P. (1982). A world championship-level othello program. *Artificial Intelligence*, 19:279–320.

Werner, G. M., and Dyer, M. G. (1991). Evolution of communication in artificial organisms. In Farmer, J. D., Langton, C., Rasmussen, S., and Taylor, C., editors, *Artificial Life II*. Reading, MA: Addison-Wesley.

Whitley, D., Starkweather, T., and Bogart, C. (1990). Genetic algorithms and neural networks: Optimizing connections and connectivity. *Parallel Computing*, 14:347–361.