TRANSNAV International Journal on Marine Navigation and Safety of Sea Transportation
Volume 6
Number 4
December 2012

Neuroevolutionary Ship Handling System in a Windy Environment

M. Łącki
Gdynia Maritime University, Gdynia, Poland

ABSTRACT: This paper presents the advanced intelligent ship handling system able to simulate and demonstrate learning behavior of artificial helmsman which controls model of ship in a windy environment of restricted water area. Simulated helmsmen are treated as individuals in population, which through environmental sensing and evolutionary algorithms learns to perform given task efficiently. The task is: safe navigation through heavy wind channels. Neuroevolutionary algorithms, which develop artificial neural networks through evolutionary operations, have been applied in this system.

1 INTRODUCTION

Decision-making processes, especially those occurring in the transport, held responsibility for the safety of people, equipment and the environment. Such important decisions should be taken with a minimum of uncertainty of the decision maker. This uncertainty may be due to the existence of a number of factors, such as: the level of training of decision-makers, the lack of information regarding the situation of the surrounding area and sometimes an excess of information provided to decision-makers simultaneously from multiple sources.

IT systems become more efficient over time regarding constant increase of computing power available to standard users. That allows developing advanced systems which collect and analyze relevant data to support decisions that minimize the risk of collision.

These advanced systems may be used in support decisions on the real ships maneuvering and ship models and simulators used during the training of future officers at training centers.

Currently still being developed computational methods are meuroevolutionary methods which, thanks to its efficiency are becoming widely used in many fields of science and technology, such as:

– automation and robotics systems, e.g. control of a robot arm (Siebel and Sommer 2007);
– designing and diagnostic systems, e.g. mobile hardware acceleration (Larkin, Kinane and O'Connor 2006), search hull damage (Kappatos, Georgoulas, Stylios and Dermatas 2009), processors design (Ratuszniak 2012), the detection and evaluation of the risk of breast cancer (Janghel, Tiwari, Kala and Shukla 2012)
– control systems: for example, a helicopter flight stabilization (De Nardi, Togelius, Holland and Lucas 2006);
– decision support systems, e.g. systems applied artificial intelligence in computer games (Kenneth, Bryant and Risto 2005).

A large number of positive results of the implementation of neuroevolutionary methods obtained in many areas of science encouraged the author of this paper to undertake research and develop his own algorithms intended for use in maritime transport (Łącki 2010b, a).

In this paper, the extension of the functionality of navigational decision support systems is proposed. This solution generates specifications of maneuvering decisions (rudder angle and propeller thrust) that maintain a safe ship trajectory computed in the available water channel. In addition to rudder angle and propeller thrust this system also includes information about time of their execution. It is possible in this system that all maneuvering decisions may be calculated and presented in real time for a given ship dynamics in the presence of certain external disturbances.

1.1 *Reinforcement Learning Algorithms*

One of the main tasks in machine learning is to create the advanced systems that can effectively find a solution of given problem and improve it over time. Reinforcement learning is a kind of machine learning, in which an autonomous unit, called a robot or agent, performs actions in a given environment. Through interaction and the observation of the environment (by input signals) and performing an action he affects this environment and receives an immediate score called reinforcement or reward (Figure 1).

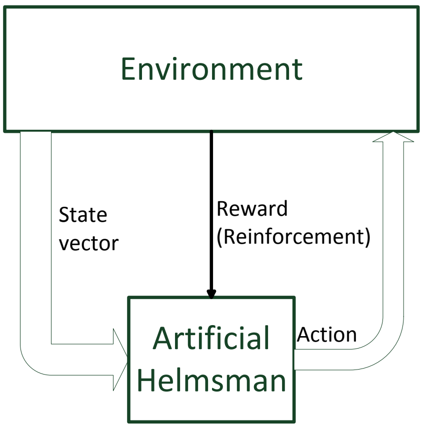

Figure 1. Interaction of helmsman with an environment.

The main task of the agent is to take such actions to adjust the value R which is the sum of the partial reinforcement as much as possible (1).

$$R_T = r_{t+1} + r_{t+2} + \cdots\ r_T \tag{1}$$

Such abilities are very important for simulating helmsman behavior in ship maneuvering on restricted waters.

For simpler layouts learning process can be performed using classic approach, i.e. Temporal Difference Reinforcement Learning (Tesauro 1995; Kaelbling, Littman and Moore 1996) or Artificial Neural Networks with fixed structures. Dealing with high-dimensional spaces is a known challenge in Reinforcement Learning approach (Łącki 2007) which predicts the long-term reward for taking actions in different states (Sutton and Barto 1998).

Evolving neural networks with genetic algorithms has been highly effective in advanced tasks, particularly those with continuous hidden states (Kenneth, Bryant and Risto 2005). Neuroevolution gives an advantage from evolving neural network topologies along with weights which can effectively store action values in machine learning tasks. The main idea of using evolutionary neural networks in ship handling is based on evolving population of helmsmen.

The artificial neural network is the helmsman's brain making him capable of observing actual navigational situation by input signals and choosing an appropriate action. These input signals are calculated and encoded from current situation of the environment.

In every time step the network calculates its output from signals received on the input layer. Output signal is then transformed into one of the available actions influencing helmsman’s environment. In this case the vessel on route within the restricted waters is part of the helmsman’s environment. Main goal of the helmsmen is to maximize their fitness values. These values are calculated from helmsmen behavior during simulation. The best-fitted individuals, which react properly to wind effect, become parents for next generation.

2 THE FORCES ACTING ON THE SHIP

External disturbances acting on a seagoing ship are mainly wind, wave and ocean current. In this paper the author assumes that the waves in the harbor area have little effect on ship maneuvering, and this type of interference is omitted in the system.

As the ship moves forward on the straight path (assuming there aren’t any significant distorting external forces) there are the two major forces acting on her - the force from the propellers and the force of water resistance. At a constant speed, these forces are equal, but with the opposite direction.

When the ship is turning the additional forces act on the rudder and lateral forces from water pressure appear. During this maneuver, the ship loses a little of her velocity and the pivot point moves back toward amidships.

2.1 *Effect of wind in the ship handling*

Under pressure of wind force, depending of the ships’ design (location of the superstructure, the deployment of cargo and on-board equipment, etc.) she tends to deviate from the course, with the wind or into the wind. The smaller the speed and draft of the ship, the greater the influence of wind. Of course, the size of the side surface exposed to wind is essential to the ships movement.

When ship moves forward the center of effort of the wind (WP) is generally close to amidships, away from pivot point (PP). This difference creates a substantial turning lever between PP and WP thus making the ship to swing of the bow into the wind (with the superstructure deployment at stern) (Figure 2).

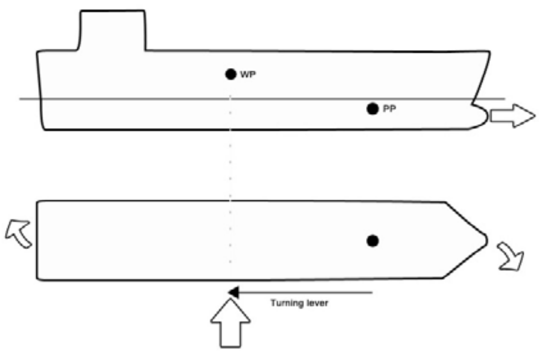

Figure 2. Wind effect and turning lever of ship mowing forward.

For ship moving forward there are defined terms of relative wind speed V_{rw} and angle of attack γ_{rw} as follows (Fossen 2011):

$$V_{rw} = \sqrt{u^2{}_{rw} + v^2{}_{rw}} \tag{2}$$

$$\gamma_{rw} = -\arctan(v_{rw,}u_{rw}) \tag{3}$$

where:

$$u_{rw} = u - u_w \tag{4}$$

$$v_{rw} = v - v_w \tag{5}$$

where: u, u_w, v, v_w are longitudinal and lateral velocities of ship and wind, respectively.

Wind forces acting on ship are generally calculated as follows:

$$X_{wind} = \frac{1}{2}\rho_{air}V_{rw}^2 C_X(\gamma_{rw})A_{Fw} \tag{6}$$

$$Y_{wind} = \frac{1}{2}\rho_{air}V_{rw}^2 C_Y(\gamma_{rw})A_{Lw} \tag{6}$$

$$N_{wind} = \frac{1}{2}\rho_{air}V_{rw}^2 C_N(\gamma_{rw})A_{Fw}L_{0a} \tag{6}$$

where:

- ρ_{air} – air density,
- A_x – surfaces affected by wind,
- L_{0a} – ship's length,
- C_n – coefficients calculated from available characteristics of ships' model (Figure 3.),

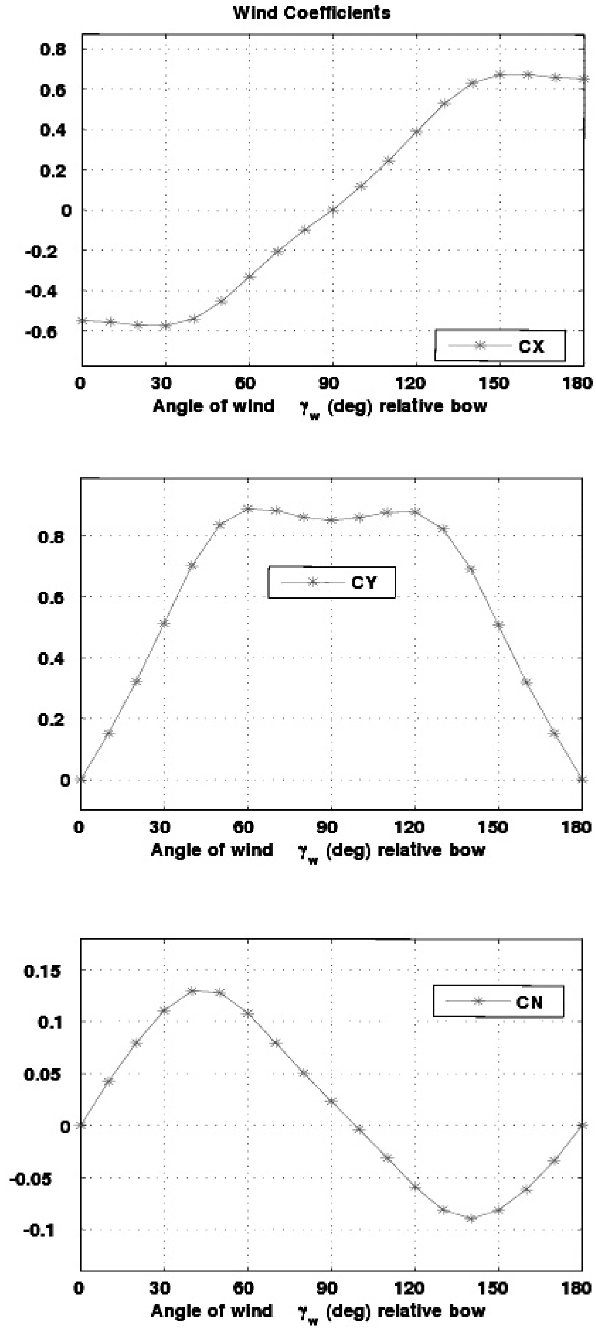

Figure 3. Equations coefficients for relative wind.

3 NEUROEVOLUTION OF AUGMENTING TOPOLOGIES

Neuroevolution of Augmenting Topologies (NEAT) method is one of the Topology and Weight Evolving Artificial Neural Networks (TWEANN's) method (Kenneth and Risto 2002b). In this method the whole population begins evolution with minimal networks structures and adds nodes and connections to them over generations, allowing complex problems to be solved gradually starting from simple ones.

The modified NEAT method consist four fundamental rules which deal with challenges that exist in evolving efficient neural network topology:

1 Begin with a minimal structure and add neurons and connections between them gradually to discover most efficient solutions throughout evolution.
2 Cross-over disparate topologies in a meaningful way by matching up genes with the same historical markings.
3 Separate each innovative individual into a different species to protect it disappearing from the population prematurely.
4 Reduce oversized topologies by removing neurons and connection between them to provide and sustain good overall performance of a whole population of helmsmen.

3.1 *Genetic Encoding*

Evolving structure requires a flexible genetic encoding. In order to allow structures to increase their complexity, their representations must be dynamic and expandable (Braun and Weisbrod 1993). Each genome in NEAT includes a number of inputs, neurons and outputs, as well as a list of connection genes, each of which refers to two nodes being connected (Figure 4.).

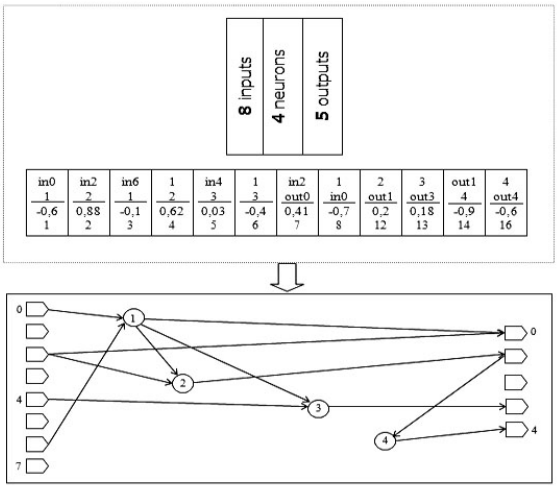

Figure 4. Genotype and phenotype of evolutionary neural network.

In this approach each connection gene specifies the output node, the input node, the weight of the connection, and an innovation number, which allows finding corresponding genes during crossover. Connection loopbacks are also allowed, as shown in figure 4.

3.2 *Genetic operations*

There are two main genetic operations: cross-over and mutation. During cross-over two individuals (parents), exchange their genetic material in purpose of creating new individual (an offspring). The system knows exactly which genes match up with which through innovation numbers. Genes that do not match are either disjoint or excess, depending on whether they occur within or outside the range of the other parent's innovation numbers.

In crossing over operation, the genes with the same innovation numbers are lined up. The offspring is then formed in one of three ways:

– In uniform crossover: matching genes are randomly chosen for the offspring genome, with all disjoints and excesses from both parents.
– In blended crossover: the connection weights of matching genes are averaged, disjoints and excesses are chosen randomly.
– In elite crossover: disjoints and excesses are taken from more fit parent only, all redundant genes from less fit parent are discarded. All matching genes are averaged.

These types of crossover were found to be most effective in evolution of neural networks in extensive testing compared to other methods of crossover (Kenneth and Risto 2002a).

Disabled genes have a chance of being re-enabled during mutation, allowing networks to make use of older genes once again.

Evolutionary neural network can keep historic trails of the origin of every gene in the population, allowing matching genes to be found and identified even in different genome structures. Old behaviors encoded in the pre-existing network structure have a chance to not to be destroyed and pass their properties through evolution to the new structures, thus provide an opportunity to elaborate on these original behaviors.

Through mutation, the genomes in modified NEAT will gradually get larger for complex tasks and lower their size in simpler ones. Genomes of varying sizes will result, sometimes with different connections at the same positions. Any crossover operator must be able to recombine networks with differing topologies, which can be difficult. Historical markings represented by innovation numbers allow NEAT to perform crossover without analyzing topologies. Genomes of different organizations and sizes stay compatible throughout evolution, and the variable-length genome problem is essentially solved. This methodology allows NEAT to increase complexity of structure while different networks still remain compatible.

Additionally different sizes and structures of networks group their genetic material into species.

3.3 *Speciation*

Speciation of the population assures that individuals compete primarily within their own niches instead of competition within the whole population. In this way topological innovations of neural network are protected and have time to optimize their structure before they have to compete with other experienced agents in the population.

During species assigning process, as described in (Łącki 2010c), when a new individual appears in population, its genome shall be assigned to one of the existing species. If this offspring is structurally too innovative comparing to any other individuals in whole population, the new species is created.

In the first step of species reproduction process the system eliminates the lowest performing members from the population. In the next step the offspring replaces eliminated worst individual (Fig. 5).

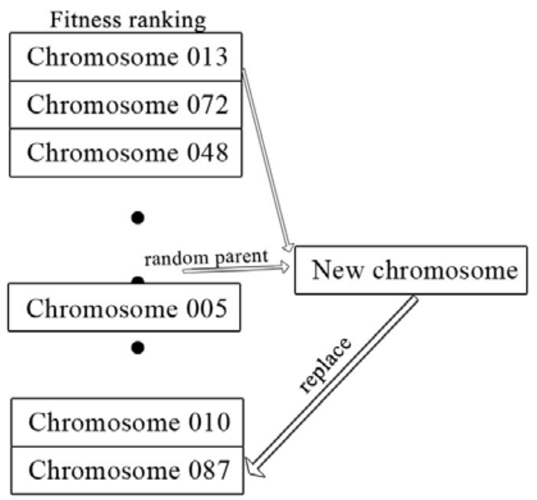

Figure 5. Example of reproduction in elitist selection method.

4 SIMULATION RESULTS

The main goal of authors work is to make a system able to simulate a safe passage of ship moving through a restricted coastal area in heavy and variable wind conditions. This goal may be achieved with Evolutionary Neural Networks.

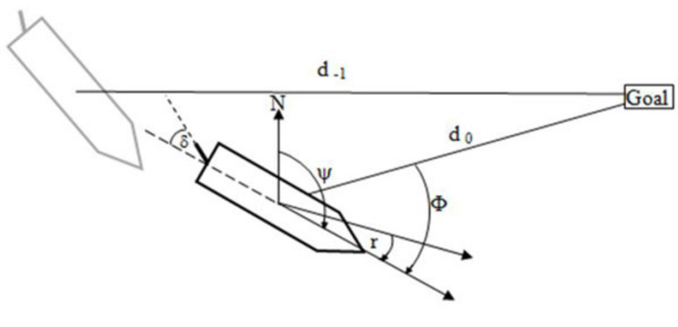

Figure 6. Sample data signals of ship handling with ENN.

Navigational situation of a moving vessel can be described in many ways. Most important is to define proper state vector from abundant range of data signals (Fig. 6.) and arbitrary determine fitness function values received by the helmsman.

The main input signals are gathered from data listed below:

- Ships course over ground,
- Ships angular velocity,
- The ship is on the collision course with an obstacle,
- Distance to collision,
- The ship is approaching destination,
- Ships angle to destination,
- The ship is heading out of the area,
- Distance to current canal borders,
- Ship is heading on goal,
- Distance to goal,
- Wind velocity,
- Angle of wind.

All the input signals are encoded either binary (0 or 1) or as a real values between 0 and 1. Some of the input signals may be calculated as multi-criteria values (Filipowicz, Łącki and Szłapczyńska 2006). Neural network output values are signals for rudder angle (δ) [deg] and thrust control [rpm].

Fitness calculation defines helmsman ability to avoid obstacles and react to wind forces while sailing toward designated goal. The fitness value of an individual is calculated from arbitrary set action values, i.e.:

- -10 when ship is on the collision course (with an obstacle or shallow waters),
- +10 when she's heading to goal without any obstacles on course,
- -100 when she hits an obstacle or run aground,
- +100 when ship reaches a goal,
- -100 when she departs from the area in any other way, etc.

In the simulation of safe passage through restricted waters there are no moving vessels in the area (Fig. 7.). In this situation when ship enters a heavy side wind channel, there is a risk to hit an obstacle if no action is being made by the helmsman. Artificial helmsman observes current situation which is encoded as input signals for his neural network and calculates the best (in his opinion) rudder angle (Figure 8).

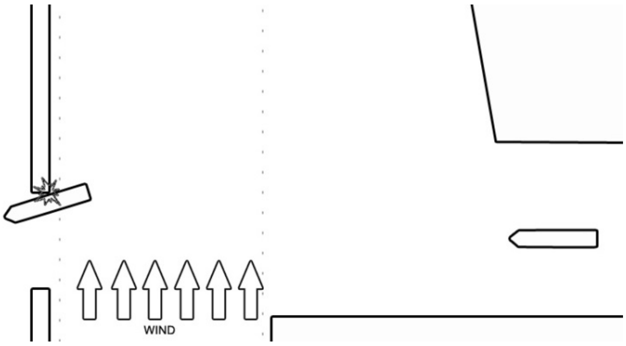

Figure 7. Model of windy coastal environment.

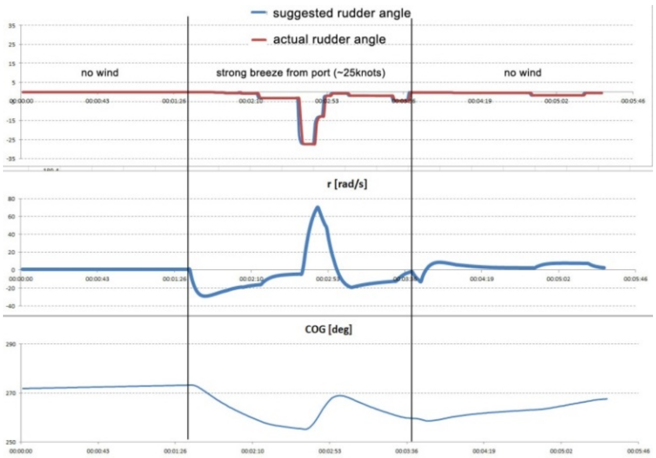

Figure 8. Simulation results of the systems performance in heavy wind environment.

5 REMARKS

Neuroevolution approach to intelligent agents training tasks can effectively improve learning process of simulated helmsman behavior in ship handling (Łącki 2008). Artificial neural networks based on NEAT increase complexity of considered model of ship maneuvering in restricted waters.

Implementation of additional disturbances from wind in neuroevolutionary system allows simulating complex behavior of the helmsman in the environments with much larger state space than it was possible in a classic state machine learning algorithms (Łącki 2007). Positive simulation results of maneuvers in variable wind conditions encourage to add other input signals to the system, like river currents, which will be included in future research.

REFERENCES

Braun H. and Weisbrod J. (1993). Evolving Feedforward Neural Networks. International Conference on Artificial Neural Networks and Genetic Algorithms, Innsbruck, Springer-Verlag.

De Nardi R., Togelius J., Holland O. E. and Lucas S. M. (2006). "Evolution of Neural Networks for Helicopter Control: Why Modularity Matters." Evolutionary Computation, 2006. CEC 2006. IEEE Congress on: 1799-1806.

Filipowicz W., Łącki M. and Szłapczyńska J. (2006). "Multicriteria Decision Support for Vessels Routing." Archives of Transport 17: 71-83.

Fossen T., I. (2011). Handbook of marine craft hydrodynamics and motion control, John Wiley & Sons, Ltd.

Janghel R. R., Tiwari R., Kala R. and Shukla A. (2012). "Breast Cancer Data Prediction by Dimensionality Reduction Using PCA and Adaptive Neuro Evolution." IJISSC 3(1): 1-9.

Kaelbling L. P., Littman M. L. and Moore A. W. (1996). "Reinforcement Learning: A Survey." Journal of Artificial Intelligence Research cs.AI/9605: 237-285.

Kappatos V. A., Georgoulas G., Stylios C. D. and Dermatas E. S. (2009). Evolutionary dimensionality reduction for crack localization in ship structures using a hybrid computational intelligent approach. Proceedings of the 2009 international joint conference on Neural Networks. Atlanta, Georgia, USA, IEEE Press: 1907-1914.

Kenneth O. S., Bryant B. D. and Risto M. (2005). "Real-time neuroevolution in the NERO video game." IEEE Transactions on Evolutionary Computation 9(6): 653-668.

Kenneth O. S. and Risto M. (2002a). Efficient evolution of neural network topologies. Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02, IEEE Computer Society: 1757-1762.

Kenneth O. S. and Risto M. (2002b). Efficient Reinforcement Learning Through Evolving Neural Network Topologies. Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers Inc.: 569-577.

Larkin D., Kinane A. and O'Connor N. (2006). Towards hardware acceleration of neuroevolution for multimedia processing applications on mobile devices. Proceedings of the 13th international conference on Neural information processing - Volume Part III. Hong Kong, China, Springer-Verlag: 1178-1188.

Łącki M. (2007). Machine Learning Algorithms in Decision Making Support in Ship Handling. TST, Katowice-Ustroń, WKŁ.

Łącki M. (2008). Neuroevolutionary approach towards ship handling. TST, Katowice-Ustroń, WKŁ.

Łącki M. (2010a). "Model środowiska wieloagentowego w neuroewolucyjnym sterowaniu statkiem." Zeszty Naukowe Akademii Morskiej w Gdyni 67: 31-37.

Łącki M. (2010b). "Wyznaczanie punktów trasy w neuroewolucyjnym sterowaniu statkiem." Logistyka 6.

Łącki M. (2010c) Speciation of Population in Neuroevolutionary Ship Handling. TransNav - International Journal on Marine Navigation and Safety of Sea Transportation, 4(2), 211-216

Ratuszniak P. (2012). "Processor array design with the use a genetic algorithm." Lecture Notes in Computer Science 7116.

Siebel N. T. and Sommer G. (2007). "Evolutionary reinforcement learning of artificial neural networks." International Journal of Hybrid Intelligent Systems - Hybridization of Intelligent Systems 4(3): 171-183.

Sutton R. and Barto A. (1998). Reinforcement Learning: An Introduction, The MIT Press.

Tesauro G. (1995). "Temporal difference learning and TD-Gammon." Communications of the ACM 38(3): 58-68