TRANSNAV International Journal on Marine Navigation and Safety of Sea Transportation

Volume 4
Number 2
June 2010

Speciation of Population in Neuroevolutionary Ship Handling

M. Lacki
Gdynia Maritime University, Gdynia, Poland

ABSTRACT: This paper presents the idea of using machine learning techniques to simulate and demonstrate learning behavior in ship maneuvering. Simulated helmsman is treated as an individual in population, which through environmental sensing learns itself to navigate through restricted waters selecting an optimum trajectory. Learning phase of the task is to observe current situation and choose one of the available actions. The individual improves his fitness function with reaching destination and decreases its value for hitting an obstacle. Neuroevolutionary approach is used to solve this task. Speciation of population is proposed as a method to secure innovative solutions.

1 ARTIFICIAL INTELIGENCE IN DECISSION MAKING SUPPORT

1.1 *Introduction*

In Artificial Intelligence (AI) one of the main tasks is to create intelligent agents that adapt to current situation, i.e. change their behavior based on interactions with the environment (Fig 1.), becoming more efficient over time, and adapting to new situations as they occur.

Such ability is important for simulating helmsman behavior in ship maneuvering on restricted waters.

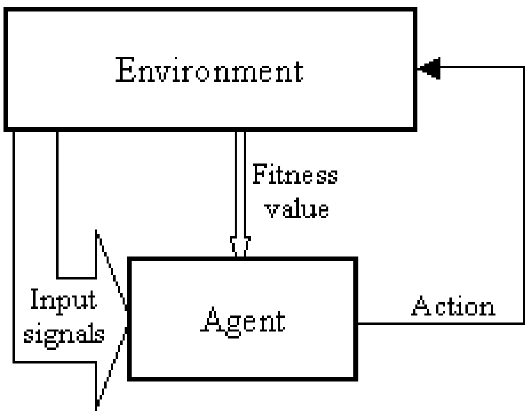

Figure 1. General model of agent-based systems.

Learning process for simpler layouts can be performed using classic approach, i.e. Temporal Difference Reinforcement Learning (Tesauro 1995) or Artificial Neural Networks with fixed structures (Braun & Weisbrod 1993). Dealing with high-dimensional spaces is a known challenge in Reinforcement Learning approach which predicts the long-term reward for taking actions in different states (Sutton & Barto 1998).

1.2 *Reinforcement Learning approach*

Reinforcement Learning algorithms were taken into consideration in previous research studies by the author (Łącki 2007). In this approach the agent receives description of current situation from the environment and chooses one of available actions. Environmental situation, which should fundamentally affect agents' behavior, is described by actual state and signal called reward. The agents' goal is to maximize total amount of reward collected over time. In simpler case total accumulated reward is a sum of immediate rewards received in every time step. Unfortunately the results of extensive simulations were insufficient in high-dimensional environment, such as helmsman behavior in ship maneuvering on restricted waters. Since simulated model of environment consist only one active agent at a time, the overall learning speed was rather slow. It has occurred that the state space was too huge to allow the agent to learn effectively. Coarse coding of states (Sutton 1996) and simplification of state vector has speeded up learning process. At the same time inaccuracy in model of restricted waters environment increases. In the long run the agent was able to take the proper action to actual task but had to learn correct behavior for slightly different task by searching whole state space again. Due to limited computer resources the state space boundaries must be defined at the beginning of simulation. Furthermore to improve state-action pair value backups in episodic learning process the eligibility traces where used, which also requires additional memory resources.

In advanced tasks, particularly those with continuous hidden states and high-dimensional spaces, evolutionary approach to artificial neural networks, has proven to be more efficient.

1.3 *Neuroevolutionary approach*

Neuroevolution is evolving neural networks, both connection weights and structure, with genetic algorithms. The main idea of using evolutionary neural networks (ENNs) in ship handling is based on training population of helmsmen (Łącki 2008).

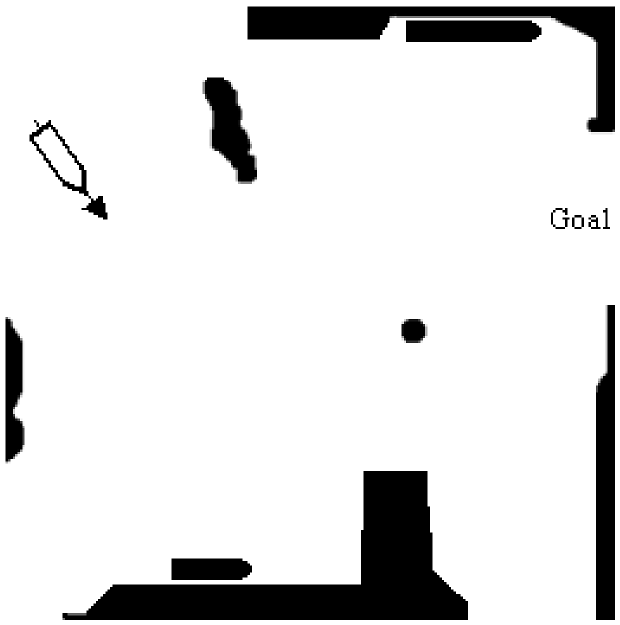

Figure 2. Model of restricted waters environment.

The neural network is the helmsman's mind allowing him to make decisions based on actual navigational situation which is represented by input signals received from environment. In each step the network calculates its output from signals received. These input signals are calculated from current situation of the environment, in this case: vessel in a confined area.

Neural network output value is the rudder angle. In actual evaluation there is only single output. Its value, which is calculated through evolution process of individuals in population, is normalized to rudder angle range from -35 degrees (port) to 35 degrees (starboard). There are plans to introduce several neural network outputs with normalization in order to bring the approach close to neural network decision support systems.

Classic artificial neural networks are not adequate in dynamic environment. Ship handling in restricted waters requires efficient network topology of helmsman's mind. To create such structure appeared to be a difficult task. The main cause of this difficulty comes from unknown hidden states and abundant variety of input signals. Furthermore evolutionary approach to neural networks is multi-agent system. It means that there are autonomous units searching for optimal solution simultaneously (Fig. 3).

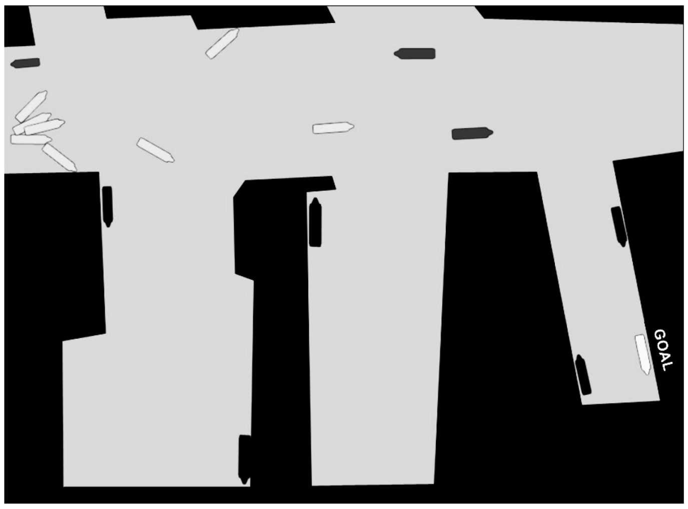

Figure 3. Multi-agent simulation system. Helmsmen compete with each other simultaneously to find the optimum route to goal.

In agent-based systems most important is to define proper state vector from available data signals derived from environment. It is also crucial to determine fitness function values received by the agent (Łącki 2008). Fitness calculation is of primary meaning when determining the quality of each individual. Subsequently it defines helmsman's ability to avoid obstacles while sailing toward designated goal.

The fitness value of an individual is adjusted in two ways: from arbitrary set action values and from calculated values, i.e.: distance to goal, relative heading to goal, distance to closest obstacle, etc.

Subjectively assigned action values are as follows: -1 if action leads to increase of the distance to goal in every time step, -10 when the 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 and –100 when she depart from the area in any other way, etc;

To simplify calculations ship's dynamic was reduced. For example speed of the ship remains constant despite significant radar deflection.

1.4 *Multi-criteria input signals*

Evaluation of quality of a state is to be treated as multi-criteria problem. Its aim is to estimate a risk factor of getting stranded, getting too close to the shore, encountering a vessel with dangerous cargo, etc. It can be estimated by function of ship's position, course and angular velocity and information gained from other vessels (if considered in the model) and coastal operators. One of the efficient methods to estimate value of risk factor is Fuzzy TOPSIS (Filipowicz, Łącki & Szłapczyńska 2005).

TOPSIS stands for Technique for Order Preference by Similarity to an Ideal Solution. Was originated by Hwang and Yoon as a new multi-attribute decision making (MADM) method in 1981. Initially the approach was intended for crisp values then extended for fuzzy parameters (Chu & Lin 2003).

The main concept of this method is based on distance calculation. The best alternative among the available set is the closest to the best possible solution and the farthest from the worst possible solution simultaneously. The best possible solution, referred to as an ideal one, is defined as a set of the best attribute values, whereas the worst possible one, referred to as a negative-ideal solution, is a set of the worst at-tribute values. In this method every criteria is of benefit or cost type. In the discussed problem distance to closest obstacle is benefit criteria (should be kept as high as possible), while probability of encountering a vessel with dangerous cargo is a cost one (therefore is to be as low as justified).

The final TOPSIS ranking is created by sorting the coefficient values assigned to each of the alternatives in descending order. The alternative with the highest ranking value claims to be the best one.

When vessels hits an obstacle or depart from the area in forbidden way then its position is reset to initial values and the helmsman receives negative points to his fitness value. The ones that reach the goal reset their positions to initial ones and increases helmsmen fitness values respectively. Therefore, after several dozen of episodes there will be some of the individuals distinguished by their high fitness values.

The main goal of the individuals in population is to maximize their fitness values. This value is calculated from helmsman behavior during simulation as described above. The best-fitted individuals become parents for next generation.

Offspring genome is calculated from parents' genomes using evolutionary operations.

2 EVOLUTIONARY OPERATIONS IN NEAT NETWORKS

Neuroevolutionary systems are based on Topology and Weight Evolving Artificial Neural Networks (TWEANNs). These neural networks have the disadvantage that the correct although simplified topology need not be known at the beginning – it will evolve through evolutionary operations.

Among TWEANNs there is Neuro Evolution of Augmenting Topologies (NEAT). It is unique in that it begins evolution with a population of minimal networks and adds nodes and connections to them over generations, allowing complex problems to be solved gradually based on simple ones (Stanley & Miikkulainen 2002). This way, NEAT searches through a minimal number of weight dimensions and finds the appropriate complexity level of network topology adjusted to the problem. This process of complexification has important implications on search patterns. It may not be practical to find a solution in a high-dimensional space by searching in that space directly. But it may be possible to find solution by searching in lower dimensional spaces and further transfer of the best solutions into the high-dimensional space.

The NEAT network delivers solutions to three fundamental problems in evolving artificial neural network topologies:

- Innovation numbers line up genes with the same origin to allow disparate topologies to cross over in a meaningful way (innovation number is a unique value assigned to a new gene).
- Separation of each innovation into a different species protects its disappearing from the population prematurely.
- Start from a minimal structure, add nodes and connections, incrementally discovers most efficient network topologies throughout evolution.

2.1 *Selection*

There are many ways to select individuals to become potential parents for next generation. Replacing the entire population on each generation may cause fast convergence to local extremes since there is strong selection method causing that everyone's genome would likely be inherited from best fitted individual. In addition, behaviors would remain static during the large gaps of time between generations.

The alternative is to replace a single individual every few time intervals as it is done in evolutionary strategy algorithms (Beyer & Schwefel 2002).

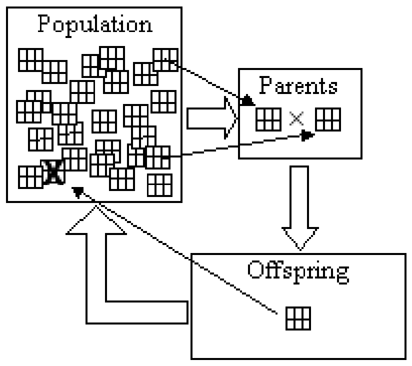

Figure 4. Evolution in population without species.

The worst individual, the one with lowest fitness value, is removed and replaced with an offspring of parents chosen from among the best. This cycle of removal and replacement happens continually throughout the simulation (Fig. 4).

2.2 *Crossover*

Every time a new connection gene appears in genome, what can only happen with mutation, a unique value is assigned to this gene called innovation number. Through innovation numbers, the system knows exactly which genes match up with another. The numbers are inherited and during crossover remain? unchanged, and allow algorithm to perform evolutionary operations without the need for expensive topological analysis. 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.

During crossover the genes with the same innovation numbers are lined up. The connection weights of matching genes are averaged.

The disjoint and excess genes are inherited from the more fit parent or, if they are equally fit, from both parents. Disabled genes have a chance of being re-enabled during crossover, allowing networks to make use of older genes once again.

2.3 *Mutation*

Mutation is the main evolving mechanism in evolutionary neural networks. It can change both network topology and connection weights.

Connection weights mutate as in any neuroevolutionary systems, with each existing connection between nodes either affected or not.

Structural mutations, which form the basis of network complexity, occur in three ways. Each mutation expands the size of the genome by adding genes. In the add connection mutation, a single new connection gene is added connecting two previously unconnected nodes. In the add node mutation, an existing connection is split and the new node placed where the old connection used to be. The old connection is disabled and two new connections are added to the genome. In the add layer mutation, a new layer is created, if the maximum layer number has not been reached yet. After that there is possibility to evolve new nodes in that new layer by add node mutation.

There are also mutations removing connections, nodes and layers and special mutation disabling particular node. This node can be re-enabled in future mutations. Probability of each type of mutation is obviously different but its value is of primary meaning in efficient evolution.

3 SPECIATION

Speciation can be seen as a result from the same process as adaptation: natural selection exerted by interaction among organisms, and between organisms and their environment. Divergent adaptation of different populations would lead to speciation.

In the course of the modern synthesis in the 20th century a somewhat different view emerged that considered speciation and divergent adaptation, the two separate processes required for the origin of species diversity, mainly as resulting from different and unrelated mechanisms. Speciation of the population assures that individuals compete primarily within their own niches instead of competition within the whole population (Stanley & Miikkulainen 2005). In this way topological innovations are protected and have time to optimize their structure before they have to compete with other niches in the population.

3.1 *Algorithm*

When new individual appears in population, it must be assigned to one of the existing species or, if it is too innovative comparing to any other individuals, new species is created. The whole species assigning algorithm is presented below.

```
Begin of the Genome Loop:
  Take the next genome $g$ from population $P$;
  Begin of the Species Loop:
    If all species in $S$ have been checked:
      create new species $s_{new}$ and place $g$ in it;
    Else
    Get the next species $s$ from $S$;
    If $g$ is compatible with $s$, add $g$ to $s$;
    If $g$ has not been placed:
            continue the Species Loop;
        Else exit the Species Loop;
    If not all genomes in $G$ have been placed:
        continue the Genome Loop;
    Else exit the Genome Loop;
```

Compatibility of genome *g* with species *s* is estimated accordingly to value of distance δ between two individuals which is calculated with formula 1:

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \overline{W} \quad (1)$$

where: c_1, c_2, c_3 – weight (importance) coefficients; E – number of excesses; D – number of disjoints; W – average weight differences of matching genes; N – the number of genes in the larger genome.

If $\delta \le \delta_t$, a compatibility threshold, then genome *g* is placed into this species.

One can avoid the problem of choosing the best value of δ by making δ_t dynamic. The algorithm can raise δ_t if there are too many species in population, and lower δ_t if there are too few.

3.2 *Fitness sharing*

Fitness sharing means that organisms in the same species must share the fitness of their niche. Thus, a species cannot afford to become too big even if many of its individuals perform well.

Therefore, any one species is unlikely to take over the entire population, which is crucial for speciated evolution to maintain topological diversity. The adjusted fitness f_i' for individual *i* is calculated according to its distance δ from every other individual *j* in the population:

$$f_i' = \frac{f_i}{\sum_{j=1}^{n} sh(\delta(i,j))} \quad (2)$$

The sharing function *sh* is set to 0 when distance δ(i,j) is above the threshold δ_t; otherwise, sh(δ(i,j)) is set to 1 (Spears 1995). Thus, sum of *sh* calculates the number of organisms in the same species as individual *i*. This reduction is natural since species are already clustered by compatibility using the threshold δ_t. A potentially different number of offspring is assigned to every species. This number is proportional to the calculated sum of adjusted fitness values f_i' of its members.

Species reproduce by first eliminating the lowest performing members from the population. In the next step the entire population is replaced by the offspring of the remaining organisms in each species (Fig. 5). The other selection methods in speciated population are also considered in future research, i.e. island selection or permanent isolation of best fitted individuals of every species with particular task.

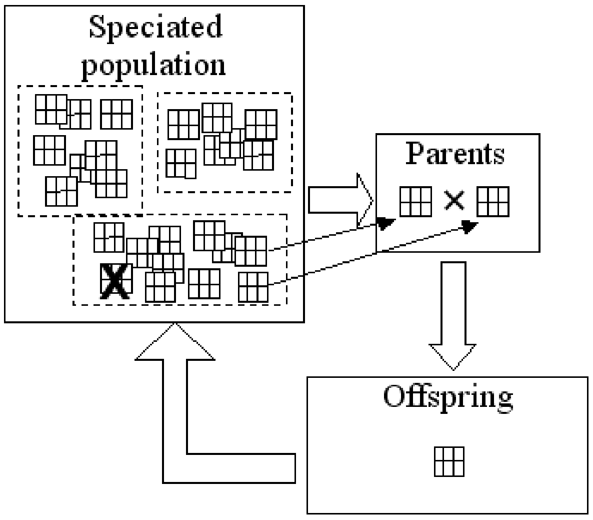

Figure 5. Evolution within one species in speciated population.

The final effect of speciating the population is that structural innovations are protected.

4 REMARKS

Speciation of population in neuroevolutionary machine learning can effectively improve learning process and decision making support in ship handling. Artificial neural networks with evolving topology and weights based on modified NEAT networks can increase learning speed of helmsmen. Complexity of considered model of ship maneuvering in restricted waters environment does not affect learning process very much. It is possible to use simulation models with much larger state space than it was possible in classic state machine learning algorithms without neural network function approximations (Kaelbling, Littman & Moore 1996). Issues like different selection methods of best fitted individuals, input signals encoding, splitting one output to several neural network outputs with normalization of signal values are also worth to be revised in future research in area of artificial intelligence support towards ship handling.

REFERENCES

Beyer, H.-G. & Paul Schwefel, H. 2002. *Evolution strategies – A comprehensive introduction*. Natural Computing, 1(1):3–52.

Braun, H. & Weisbrod, J. 1993. *Evolving feedforward neural networks*. Proceedings of ANNGA93, International Conference on Artificial Neural Networks and Genetic Algorithms. Berlin: Springer.

Chu T. C., Lin Y. C. 2003. *A Fuzzy TOPSIS Method for Robot Selection*, the International Journal of Advanced Manufacturing Technology: 284-290,

Filipowicz, W., Łącki, M. & Szłapczyńska, J. 2005, *Multicriteria decision support for vessels routing,* Proceedings of ESREL'05 Conference.

Kaelbling, L. P., Littman & Moore. 1996. *Reinforcement Learning: A Survey.*

Łącki M., 2007 *Machine Learning Algorithms in Decision Making Support in Ship Handling*, Proceedings of TST Conference, Katowice-Ustroń, WKŁ.

Łącki, M. 2008, *Neuroevolutionary approach towards ship handling*, Proceedings of TST Conference, Katowice-Ustroń, WKŁ.

Spears, W. 1995. *Speciation using tag bits*. Handbook of Evolutionary Computation. IOP Publishing Ltd. and Oxford University Press.

Stanley, K. O. & Miikkulainen, R. 2002. *Efficient reinforcement learning through evolving neural network topologies*. Proceedings of the Genetic and Evolutionary Computation. Conference (GECCO-2002). San Francisco, CA: Morgan Kaufmann.

Stanley, K. O. & Miikkulainen, R. 2005. *Real-Time Neuroevolution in the NERO Video Game*, Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games, Piscataway

Sutton, R. 1996. *Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding*. Touretzky, D., Mozer, M., & Hasselmo, M. (Eds.), Neural Information Processing Systems 8.

Sutton, R. & Barto, A. 1998. *Reinforcement Learning: An Introduction.*

Tesauro, G. 1995. *Temporal Difference Learning and TD-Gammon*, Communications of the Association for Computing Machinery, vol. 38, No. 3.