Analyzing Evolved Fault-Tolerant Neurocontrollers

Alon Keinan
School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel
keinanak@post.tau.ac.il

Abstract

Evolutionary autonomous agents whose behavior is determined by a neurocontroller "brain" are a promising model for studying neural processing. Nevertheless, they are missing an important quality prevalently found in all levels of natural systems, *fault-tolerance*, the lack of which results in overly simplistic neurocontrollers. We present a way of modifying a given evolutionary process for encouraging the creation of neurocontrollers that manifest high levels of fault-tolerance, using both direct and incremental evolutions. The evolved neurocontrollers are more robust not only against the faults introduced during the evolutionary process, but also against much more extreme ones. This robustness poses a great challenge for an analysis of the workings of the neurocontrollers, the latter being the focus of this paper: We utilize the Multi-perturbation Shapley value Analysis (MSA) to uncover the important neurons, as well as the interactions between them, revealing the mechanisms underlying the evolved fault-tolerance.

Introduction

Neurally-driven Evolved Autonomous Agents (EAAs) are software programs embedded in a simulated virtual environment, performing tasks such as navigating and gathering food. An agent's behavior is determined by a neurocontroller which receives and processes sensory inputs from the surrounding environment and governs the activation of its motors. In recent years, much progress has been made in finding ways to evolve agents that successfully cope with diverse behavioral tasks (see Kodjabachian and Meyer (1998); Yao (1999); Guillot and Meyer (2001) for reviews). Furthermore, numerous EAA studies have yielded networks which manifest interesting biological-like characteristics (e.g. Cangelosi and Parisi (1997); Ijspeert et al. (1999); Aharonov-Barki et al. (2001)). Hence, EAAs are a very promising model for studying neural processing and developing methods for its analysis (Ruppin, 2002). Being abstractions, EAAs are missing many qualities of natural systems. This paper focuses on one such important quality, *fault-tolerance*, which is prevalently found in all levels of living organisms. Fault-tolerance emerges as a consequence of evolutionary pressure which favors organisms which are more resilient to harm. In this paper we emulate such a pressure by modifying a given evolutionary fitness function, encouraging the creation of neurocontrollers that manifest high levels of fault-tolerance. Those neurocontrollers are more plausible, both biologically and for hardware implementation.

A large amount of work has been invested in enhancing the fault-tolerance of feedforward neural networks, either by explicitly adding redundant hidden nodes or by modifying the back-propagation training algorithm. Evolutionary techniques for producing fault-tolerant systems have been mainly developed in the context of evolvable hardware, starting from Thompson (1996) who suggested to deliberately subject an evolving system to faults during its fitness evaluations. Canham and Tyrrell (2002) have used Thompson's method, while investigating what causes an evolved circuit to be tolerant to faults. In a recent paper (Zhou and Chen, 2003), a genetic algorithm based method is applied for improving the fault-tolerance of feedforward, back-propagation trained, neural networks. All these studies first produce a processing system that perform the required function and only then aim at producing fault-tolerance. In this paper, we evolve fault-tolerant embedded neurocontrollers both using a direct evolution and an incremental one. Further, we deal with fully-recurrent networks, without an explicit error function, rather than restricting the scope to feedforward networks trained via back-propagation.

One of the major challenges in the field of EAAs is understanding the way neurocontrollers operate. Lesion studies, where functional performance is measured after lesioning different elements of the system, have been employed in neuroscience for localizing function in a causal manner. However, most of the lesion studies employ single-lesions, in which only one element is lesioned at a time. Such approaches are limited in their ability to reveal the significance of interacting elements. One obvious example is provided by two elements that exhibit a high degree of overlap in their function, as is likely to be the case in fault-tolerant systems: Lesioning either element alone will not reveal its significance. Acknowledging that single lesions are insufficient

for localizing functions in neural systems, we have previously presented the Multi-perturbation Shapley value Analysis (MSA) (Keinan et al., 2004b). The MSA processes a data set composed of numerous multiple lesions that are afflicted upon a neural system, together with the corresponding system performance level in each. It quantifies the contribution of the different system elements to the successful performance of the system's functions, as well as the functional interactions between groups of elements. The MSA was first developed for deciphering the mechanisms underlying EAAs' behavior. In comparison to previous analyzed neurocontrollers (Keinan et al., 2004a; Saggie et al., 2003; Ganon et al., 2003), the fault-tolerant agents evolved in this study pose a much greater analysis challenge, serving as a more biologically plausible testbed for such analysis methods. In this paper, we utilize the MSA for the analysis of the evolved neurocontrollers, examining the fault-tolerance mechanisms. We also present a new variant, *MSA K-limited contributions*, introduced in this paper in order to examine, in the presence of high levels of fault-tolerance, the gap between the single-lesion approach and the full multi-lesion analysis.

The Evolutionary Environment

The EAA environment is described in detail in Aharonov-Barki et al. (2001). The agents live in a discrete 2D grid "world" surrounded by walls. Poison items are scattered all around the world, while food items are scattered only in a "food zone" in one corner. The agent's goal is to find and eat as many food items as possible during its life, while avoiding the poison items. The agent is equipped with a set of sensors, motors, and a fully recurrent neurocontroller containing n McCulloch-Pitts neurons ($n = 10$ in all simulations). The four sensors encode the presence of a wall, a resource (food or poison, without distinction between the two), or a vacancy in the cell the agent occupies and in the three cells directly in front of it. A fifth sensor is a "smell" sensor which can differentiate between food and poison underneath the agent and gives a random reading if the agent is in an empty cell. The four motor neurons dictate movement forward (neuron 1), a turn left (neuron 2) or right (neuron 3), and control the state of the mouth (open or closed, neuron 4).

As in previous studies (Aharonov-Barki et al., 2001), a *genetic algorithm* is used to evolve the synaptic weights by directly encoding them in the genome as real valued numbers. In this study, in order to encourage the creation of fault-tolerant neurocontrollers, phenotypic faults are introduced while an agent's fitness is being evaluated throughout the evolutionary process, thus making the fault-tolerance an integral part of the task specification. Particularly, the fitness of an agent is determined by evaluating it n times, in addition to the regular evaluation, with a different neuron being lesioned (removed from the neurocontroller) in each of these evaluations.[1] The *mean lesion fitness* function is then defined by

$$\frac{1}{n+1}\left(f(N)+\sum_{i=1}^{n} f(N\setminus\{i\})\right), \tag{1}$$

where $N=\{1,\ldots,n\}$ is the set of neurons and f is the performance. $f(N)$ denotes the performance level of the intact neurocontroller, which is the *standard fitness* used to evolve the agents in Aharonov-Barki et al. (2001), while $f(N\setminus\{i\})$ denotes the performance level of the neurocontroller when neuron i is lesioned. The mean lesion fitness function is utilized to create fault-tolerant neurocontrollers in two ways: One is incremental evolution, in which an evolutionary run is first conducted using the standard fitness function, starting from random neurocontrollers. Then, in the incremental stage, starting from the most successful agent in this evolutionary run, agents are evolved using the mean lesion fitness function. The second one consists of a direct evolutionary run using the mean lesion fitness function, starting with random neurocontrollers. Ten evolutionary runs of each of the two types were performed, all with the same genetic algorithm parameters as in Aharonov-Barki et al. (2001).

Analysis of Evolutionary Results

We begin by testing whether the introduced change to the fitness function results in more fault-tolerant agents. Figure 1 presents both the standard and mean lesion fitness of the successful agents. The difference between the two fitness functions serves to quantify the level of fault-tolerance. As evident, the agents evolved using the mean lesion fitness are much more fault-tolerant compared with the agents evolved using the standard fitness (the ones serving as the basis for the incremental stage), while reaching almost the same level of intact performance. This testifies that **the modified evolutionary pressure indeed encourages the creation of fault-tolerance, when using either direct or incremental evolutions.**

In order to further quantify the level of fault-tolerance of a neurocontroller, we measure the degradation in the agent's performance level as it is subjected to more and more concurrent lesions. Figure 2 shows the average performance level of agent F, an incrementally evolved fault-tolerant agent, as a function of the lesioning depth (the number of concurrently lesioned neurons). Obviously, the agent after the incremental evolution maintains, for all lesion depths, a much higher level of performance than its predecessor, before applying the incremental stage of the evolution. Interestingly, **though during the evolution only**

[1] When lesioning motor neurons the activity transmitted to the motors themselves in not altered, but only the activity transmitted to other neurons, thus hindering only the role they play in the recurrent neurocontroller's computations.

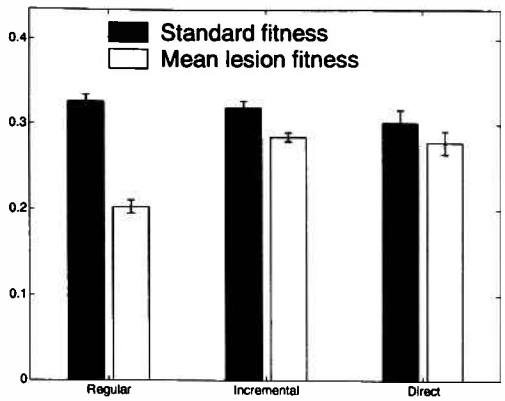

Figure 1: The standard fitness and mean lesion fitness for the agents evolved with the standard fitness (*Regular*), for the agents evolved using incremental evolution (*Incremental*) and for the agents evolved using direct evolution (*Direct*). All are mean (and standard deviation of the mean) of the most successful agent from each of the 10 evolutionary runs of that type.

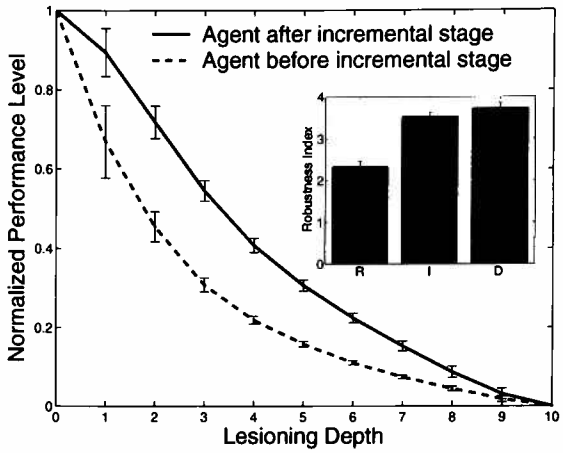

Figure 2: The mean normalized performance level as a function of the lesioning depth, for agent F (solid line) and for its predecessor, serving as the basis for the incremental stage in which F was evolved (dashed line). The robustness index equals 3.87 for the former and 2.56 for the latter. The inset depicts the mean robustness index of the agents of each type, in the same order of bars as in Figure 1.

single-lesions are afflicted, the evolutionary pressure encourages a higher level of fault-tolerance, e.g., the agent's performance level when four of its neurons are lesioned is about the same as the performance level of the agent evolved without the mean lesion fitness when only two of its neurons are lesioned. We define the *robustness index* to be the area below the curve of the type introduced in Figure 2. Evidently, based on this measure, **the evolution with the mean lesion fitness yields agents which are much more robust** (Figure 2).

Lesioning a neuron that plays no part in carrying out the neurocontroller's function might have no effect on the performance level. Hence, as suggested by Canham and Tyrrell (2002) in the context of evolvable hardware, systems whose function is carried out by a small fraction of its elements would be regarded as more robust. We wish to test whether the above results are indeed a manifestation of the evolution of backup mechanisms or whether they are merely an outcome of smaller effective sizes of the neurocontrollers evolved with the mean lesion fitness. An intuitive way to attack this question, analogous to the one suggested by Canham and Tyrrell (2002), is to test, for a neuron that has no effect on the performance level when single-lesioned, whether is has an effect when lesioned together with any other neuron. Indeed, there are incidents where this suffices to expose the effect of neurons. For instance, in agent F, lesioning either neuron number 5 or neuron number 10 by themselves does not change the performance level considerably, while lesioning them together does cause a large decrease of 62%. Nevertheless, since the evolved neurocontrollers exhibit an improved fault-tolerance to deep levels of lesioning, as shown above, double-lesions might not be enough to reveal all the neurons playing a true part in carrying out the function. Indeed, Canham and Tyrrell (2002) report that using this kind of approach still gives only an impression of what is happening, while missing elements which must be of importance. To overcome the drawbacks of finding the important neurons based on single-lesions and/or double-lesions, we turn to utilize the MSA.

The MSA framework (Keinan et al., 2004b) addresses the challenge of defining and calculating the contributions of system elements from a data set of multiple lesions that are afflicted upon the system. In this framework, we view a set of multiple lesion experiments as a *coalitional game*, borrowing concepts and analytical approaches from the field of Game Theory. Specifically, we define the set of contributions to be the *Shapley value* (Shapley, 1953), which stands for the unique fair division of the game's worth (the system's performance level when all elements are intact) among the different players (the system elements). Applying the MSA for the analysis of the neurocontrollers presented in this paper, the *marginal importance* of neuron i to a set of neurons S, with $i \notin S$, is defined by

$$\Delta_i(S) = f(S \cup \{i\}) - f(S), \tag{2}$$

where $f(S)$ is the performance level when only the neurons in the set S are intact, while the rest are lesioned. Then, the Shapley value defines the contribution

$$\gamma_i = \frac{1}{n!} \sum_{R \in \mathcal{R}} \Delta_i(S_i(R)) \tag{3}$$

of each neuron $i \in N$, where $\mathcal{R}$ is the set of all $n!$ orderings of the set of neurons N and $S_i(R)$ is the set of neurons preceding i in the ordering R. It can be interpreted as follows:

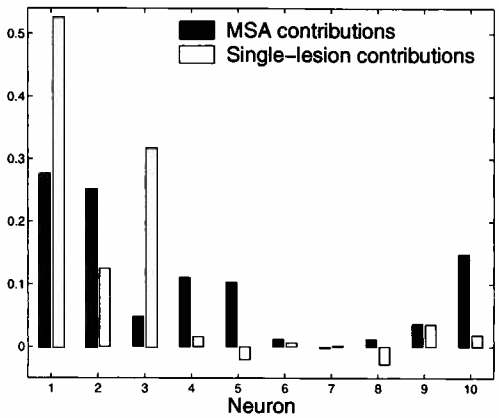

Figure 3: MSA contributions vs. single-lesion contributions of agent *F*. The single-lesion contribution of a neuron is the decrease in the performance level when the neuron is lesioned. In order to be comparable, both are normalized such that the sum across all neurons equals 1.

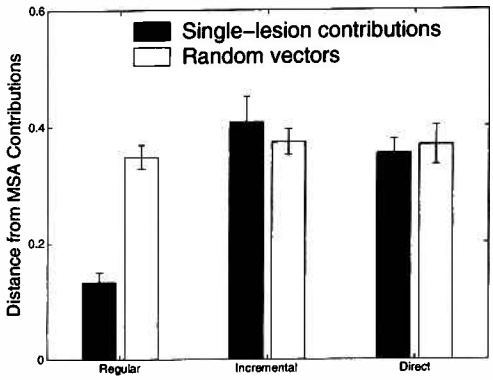

Figure 4: The Euclidean distance between the vector of normalized single-lesion contributions and the vector of normalized MSA contributions (dark bars). For comparison, the distance between random normalized vectors and the MSA contributions is also plotted (light bars). Both present the mean of all agents of each type.

Suppose that all the neurons are arranged in some order, all orders being equally likely. Then γ_i is the expected marginal importance of neuron i to the set of neurons who precede him. *This unique and fair contribution measures the part the neuron plays in successfully performing the neurocontroller's function.*

Based on all the possible multi-lesion experiments, the MSA reveals the true contribution of neurons whose importance has been missed by the single-lesion approach (Figure 3). For instance, arbitrarily defining an important neuron as one with a normalized contribution greater then 0.03, the MSA reveals 7 important neurons while the single-lesion approach reveals only 4 of them. The performance level even slightly increases when lesioning neuron number 5, resulting in a negative single-lesion contribution, while actually this neuron is an important one. Generally, **the disagreement between the single-lesion approach and the MSA is greater for the agents evolved using the mean lesion fitness, with single-lesion contributions being as far from the MSA contributions as random normalized vectors are** (Figure 4). Armed with the MSA contributions, we can finally return to test whether the neurocontrollers evolved with the mean lesion fitness are of smaller effective sizes. The results testify that this in not the case as agents of all three evolutionary types have, on the average, the same number of important neurons (mean of 7.1 neurons), with no significant difference.[2] This new finding leads to the conclusion that the agents evolved with the mean lesion fitness are **more robust due to the evolution of some backup mechanisms, rather than the mere evolution of smaller solutions**. We return to further investigate these mechanisms in the next section.

[2]This conclusion still holds for other importance thresholds, though the number of important neurons varies (data not shown).

To examine the wide gap between the contributions yielded by single-lesions and the contributions yielded by the MSA, the approach should be generalized. We seek a variant of the MSA which, for a given lesioning depth K, yields the neuronal contributions based on all multi-lesions up to that depth. Hence, we define $\Delta_i(S)$ (eq. (2)) to be zero for all S such that $|S| \geq K$. Based on this definition, the contribution of neuron i may be defined according to eq. (3), properly normalized by changing the denominator to be $K \cdot (n-1)!$ instead of $n!$. These *MSA K-limited contributions* coincide with the actual MSA contributions for $K = N$ and with the single-lesion contributions for $K = 1$. Figure 5 plots the distance between the MSA K-limited contributions and the actual ones as a function of K. Obviously, **the K-limited contributions approach the actual contributions much slower for the fault-tolerant agent, testifying that one must use a deep level of multi-lesions in order to gain true insights into the working of such systems.** For instance, an analysis of the regular neurocontroller based on single-lesions solely is as accurate as an analysis of the fault-tolerant neurocontroller based on all multi-lesions of up to depth 4.

Underlying Fault-Tolerance Mechanisms

The MSA contributions introduced in the previous section serve as a summary of the system's functionality, indicating the average marginal importance of each element over all possible orderings of the elements. For complex systems, where the importance of an element strongly depends on the state (lesioned or intact) of other elements, the MSA further suggests a higher order description (Keinan et al., 2004b). Focusing here on a two-dimensional description, the MSA defines the functional interaction between each pair of elements. This functional interaction quantifies how much the

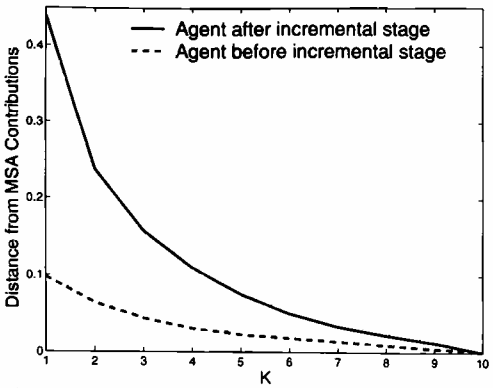

Figure 5: The Euclidean distance between the normalized MSA contributions and the normalized MSA K-limited contributions as a function of the lesioning depth K, for agent F (solid line) and for its predecessor (dashed line).

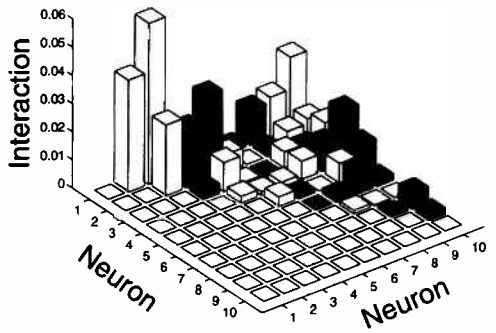

Figure 6: The symmetric MSA interactions between each pair of neurons of agent F. The figure presents the absolute values, where dark bars denote negative interactions and light bars denote positive ones.

average marginal importance of the two elements together is larger (or smaller) than the sum of the average marginal importance of each of them when the other one is lesioned. Intuitively, this interaction measures how much "the whole is greater than the sum of its parts" (*synergism*), where the whole is the pair of elements. In cases where the whole is smaller than the sum of its parts, that is, when the two elements exhibit functional overlap (*antagonism*), the interaction is negative. Clearly, single-lesion approaches cannot uncover such functional interactions.

Observing the MSA interactions between all pairs of neurons of agent F reveals many negative ones, pointing to pairs of neurons which backup each other's function (Figure 6). Obviously, the backup scenario is not a clear-cut case as one might have expected, according to which each redundant neuron has another redundant one completely backing it up. Rather, several neurons backup several others to some extent, e.g., each of neurons 4, 9 and 10 backup each of the two others. These results exemplify the multiplicity of negative interactions in the agents evolved with the mean lesion fitness. While those agents have, on the average, 15.1 negative interactions of meaningful magnitude, the agents evolved with the standard fitness have only 7.8, with this difference being significant (p-value < 0.05). Comparing the number of positive interactions, the former have much less than the latter (mean of 17.8, compared with mean of 27.1; p-value < 0.05). These results testify to the fact that **the evolutionary pressure introduced by the mean lesion fitness function encourages the formation of functional overlap between the neurons, at the expense of the formation of cooperation**, which results in the evolution of neurocontrollers that are much more fault-tolerant.

Since, as shown above, the backup scenario is not a clear-cut case, we turn to more closely analyze the fault-tolerance mechanisms by focusing on the level of the individual synaptic connections. The Evolutionary Network Minimization (ENM) algorithm (Ganon et al., 2003) is a genetic algorithm with an additional step in which synaptic connections are irreversibly eliminated. Ganon et al. (2003) have shown that, given an agent, the ENM tends to largely minimize its neurocontroller, while keeping the performance level high and the principal functional characteristics intact. Here, we utilize the ENM to minimize the successful neurocontrollers, while examining the number of remaining synaptic connections in the resulted backbones. The agents evolved with the mean lesion fitness have more synaptic connections in their backbones (mean of 20.4, compared with mean of 17.1; p-value < 0.05), implying that **those fault-tolerant agents perform a more complicated function**. Bearing in mind that, as shown in the previous section, both the fault-tolerant and the regular neurocontrollers consist of the same number of important neurons, we conjecture that the additional 3.3 synapses found, on the average, are due to synaptic connections playing a role in the backup mechanisms between those neurons. A further direct inspection of the small backbones yielded by the ENM helps in understanding the way the neurocontrollers operate and the redundancies between synapses (omitted due to space considerations).

Discussion

We have introduced a modification to evolutionary processes for evolving neurocontrollers that, while reaching a good performance level, exhibit high fault-tolerance to neuronal lesioning. The modification was shown to be successful both when starting from a successful, but not fault-tolerant, neurocontroller (incremental evolution) and when starting from random ones (direct evolution). The evolved neurocontrollers exhibit a high level of fault-tolerance, not only to the faults introduced during the evolutionary process, but also to much more extreme ones. Aiming to understand the workings of those evolved neurocontrollers, their robustness poses a great challenge. We have utilized the MSA to uncover the important neurons, as well as the interactions be-

tween them, while overcoming the inherent disadvantages of single-lesion approaches, showing that these are amplified in the face of fault-tolerance. Based on the MSA, the depth of lesioning required to get a good insight into the workings of a neurocontroller has been quantified. Furthermore, the analysis reveals that the robustness of the neurocontrollers is due to the actual evolution of backup mechanisms, captured by the functional interactions, rather than the evolution of smaller solutions. Lastly, while the fault-tolerant neurocontrollers utilize the same number of important neurons to perform their function, the underlying synaptic networks, captured by the ENM, are more complicated than the ones of regular neurocontrollers.

The development and study of the MSA has been first done within the framework of EAAs. Nevertheless, the MSA is geared toward general experimental biological applications. In Keinan et al. (2004b), the applicability of the MSA to the analysis of neurophysiological models was demonstrated, as well as to the analysis of behavioral data from experimental deactivation studies of the cat brain. The MSA is further applicable to the analysis of Transcranial Magnetic Stimulation (TMS) experiments. In biology in general, the recent development of RNA interference (RNAi) has made the possibility of multiple concomitant gene knockouts a reality, allowing the utilization of the MSA to the analysis of genetic and metabolic networks. Surely, in such biological applications only a limited number of multi-perturbation experiments can be performed. Hence, the MSA encompasses prediction and estimation variants which approximates the Shapley value in an accurate, scalable and efficient manner (Keinan et al., 2004b). Robustness, being a common quality of biological systems, must be correctly handled by any analysis method applied to biological experiments. This paper has demonstrated the applicability of the MSA to the analysis of robust systems, overcoming the disadvantages of the single-lesion approach, and has established that a deep level of lesioning should be used in order to correctly identify the important elements in such systems.

References

Aharonov-Barki, R., Beker, T., and Ruppin, E. (2001). Emergence of memory-driven command neurons in evolved artificial agents. *Neural Computation*, 13(3):691–716.

Cangelosi, A. and Parisi, D. (1997). A neural network model of Caenorhabditis Elegans: The circuit of touch sensitivity. *Neural Processing Letters*, 6:91–98.

Canham, R. O. and Tyrrell, A. M. (2002). Evolved fault tolerance in evolvable hardware. In *proceedings of the Congress on Evolutionary Computation 2002 (CEC2002)*, pages 1267–1272.

Ganon, Z., Keinan, A., and Ruppin, E. (2003). Evolutionary network minimization: Adaptive implicit pruning of successful agents. In Banzhaf, W., Christaller, T., Dittrich, P., Kim, J. T., and Ziegler, J., editors, *Advances in Artificial Life - Proceedings of the 7th European Conference on Artificial Life (ECAL)*, volume 2801 of *Lecture Notes in Artificial Intelligence*, pages 319–327. Springer Verlag Berlin, Heidelberg.

Guillot, A. and Meyer, J. A. (2001). The animat contribution to cognitive systems research. *Journal of Cognitive Systems Research*, 2(2):157–165.

Ijspeert, A. J., Hallam, J., and Willshaw, D. (1999). Evolving swimming controllers for a simulated lamprey with inspiration from neurobiology. *Adaptive Behavior*, 7(2):151–172.

Keinan, A., Hilgetag, C. C., Meilijson, I., and Ruppin, E. (2004a). Causal localization of neural function: The Shapley value method. *Neurocomputing, to appear.*

Keinan, A., Sandbank, B., Hilgetag, C. C., Meilijson, I., and Ruppin, E. (2004b). Fair attribution of functional contribution in artificial and biological networks. *Neural Computation, to appear.*

Kodjabachian, J. and Meyer, J. A. (1998). Evolution and development of neural controllers for locomotion, gradient-following and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks*, 9(5):796–812.

Ruppin, E. (2002). Evolutionary autonomous agents: A neuroscience perspective. *Nature Reviews Neuroscience*, 3:132–141.

Saggie, K., Keinan, A., and Ruppin, E. (2003). Solving a delayed response task with spiking and Mccullloch-Pitts agents. In Banzhaf, W., Christaller, T., Dittrich, P., Kim, J. T., and Ziegler, J., editors, *Advances in Artificial Life - Proceedings of the 7th European Conference on Artificial Life (ECAL)*, volume 2801 of *Lecture Notes in Artificial Intelligence*, pages 199–208. Springer Verlag Berlin, Heidelberg.

Shapley, L. S. (1953). A value for n-person games. In Kuhn, H. W. and Tucker, A. W., editors, *Contributions to the Theory of Games*, volume II of *Annals of Mathematics Studies 28*, pages 307–317. Princeton University Press, Princeton.

Thompson, A. (1996). Evolutionary techniques for fault tolerance. In *Proceedings UKACC Int. Conf. on Control (CONTROL 96)*, pages 693–698. IEE Conference Publication No. 427.

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447.

Zhou, Z. H. and Chen, S. F. (2003). Evolving fault-tolerant neural networks. *Neural Computing and Applications, 11(3-4): 156-160.*