ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/327344855

Watershed management using neuroevolution

Article · August 2018

DOI: 10.1007/s40808-018-0508-z

CITATIONS
2

READS
52

3 authors, including:

Karl Mason
Cardiff University

28 PUBLICATIONS **256** CITATIONS

SEE PROFILE

Jim Duggan
National University of Ireland, Galway

145 PUBLICATIONS **1,288** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

Digital Health View project

All content following this page was uploaded by Karl Mason on 11 January 2019.

The user has requested enhancement of the downloaded file.

Noname manuscript No.
(will be inserted by the editor)

Watershed Management using Neuroevolution

Karl Mason · Jim Duggan · Enda Howley

Received: date / Accepted: date

Abstract Neuroevolution refers to evolving neural networks using evolutionary methods. These algorithms have been applied to many problem domains, from game playing to robotics, which motivates this research. The problem of watershed management is addressed here in this research using the most prominent neuroevolution algorithms, i.e. NeuroEvolution of Augmenting Topologies (NEAT), Neuro Differential Evolution (NDE) and Enforced SubPopulations (ESP). The results indicate that neuroevolution is a suitable approach at addressing the watershed management problem, outperforming the other methods of neural network training.

Keywords Neuroevolution · Watershed Management · Control · Neural Networks · Evolutionary Algorithms

1 Introduction

Evolutionary neural networks (commonly referred to as Neuroevolution (NE)) combines two well known research areas: evolutionary algorithms and neural networks (Floreano et al. (2008)).There are many advantages to using these methods such as: there is no need for target outputs, they are suitable for large and complex problems and they are resistant to noise. This makes them particularly well suited to engineering control problems. This research will apply a range of neuroevolution algorithms to watershed management (Yang et al. (2009)). The watershed management problem consists of managing the distribution of a finite amount of water to a number of interested individuals that seek to maximise their consumption. The various uses of the water include municipal use, irrigation, hydroelectricity and for the surrounding ecosystems. There have been a number of approaches to this problem including: Multi Agent Systems (Yang et al. (2009)), Multi Agent Systems combined with Genetic Algorithm (Barbalios and Tzionas (2014)), Robust Decision Making (Kasprzyk et al. (2013)) and Multi Population Evolutionary Algorithm (Erfani and Erfani (2015)). Each of these approaches address the watershed management as a distributed control problem where multiple controllers are implemented to distribute the water to each party. This research will involve evolving a single centralized neural network controller to allocate the water to the various parties seeking water. Previous similar research has explored optimising the network's parameters using Particle Swarm Optimisation (PSO) (Mason et al. (2018)). This research is built upon by applying the most prominent neuroevolution methods to optimise both the network's topology and also its weights, i.e. NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen (2002)), Enforced SubPopulations (ESP) (Gomez and Miikkulainen (1999)) and Neuro Differential Evolution (NDE) (Mason et al. (2017)) to the task of watershed management. There are currently no applications of neuroevolution to watershed management in the literature. This research paper investigates the following:

1. To apply neuroevolution methods to watershed management.
2. To establish if evolving both the network's connectivity and its weights leads to enhanced performance, than solely optimising the network weights.

K. Mason
E-mail: karljmason@hotmail.com
· J. Duggan
E-mail: james.duggan@nuigalway.ie
· E. Howley
E-mail: ehowley@nuigalway.ie

Discipline of Information Technology,
National University of Ireland Galway

3. To compare state of the art neuroevolution methods when addressing the problem of watershed management.

2 Neuroevolution

Neuroevolution is a term used to define the use evolutionary algorithms to search for the optimal neural network configuration to approximate some function. There are no existing applications in the literature relating to the application of neuroevolution algorithms to watershed management. This is one of the primary contributions of this research.

Neuroevolution methods consist of a population of solutions. The population of networks iteratively improves over time as a result of the application of evolutionary operators to the population, such as selection, crossover and mutation. In order for neuroevolution to operate, aspects of the neural network design are encoded into genotype. These typically contain network information such as synaptic weight values, number of neurons, connectivity, etc. These network traits form the genotype. These genotypes are evolved over a series of generations. The phenotype is the expression of the genotype. In the context of neuroevolution, the phenotype is the actual neural network. At each iteration, genotypes are selected via one of many methods (linear ranking, roulette wheel, etc.). These genotypes are then mated through crossover and then possibly mutated to form the next generation of genotypes.

2.1 NEAT Algorithm

NEAT is one of the most popular neuroevolution algorithm and was first proposed by Stanley and Miikkulainen in 2002 (Stanley and Miikkulainen (2002)). It is one of the most widely used neuroevolution algorithms with many applications. For this reason, the NEAT algorithm will be one of the algorithms applied to the watershed management problem. This method uses a evolutionary operators to optimise the network's topology and also the synaptic weights. One of the key features of the NEAT algorithm is that it is initialized to produce the smallest possible network. The network parameters are mapped directly onto the genotype. As the algorithm runs, the it evaluates more complex networks with more hidden neurons. The genes are categorized into species based on their innovation number. NEAT remembers the innovation number of each gene so that crossover can only occur between networks within the same species.

2.2 ESP Algorithm

The second algorithm that will be implemented is Enforced SubPopulations (ESP) (Gomez and Miikkulainen (1999)). Unlike NEAT, ESP evolves a network with a constant number of hidden neurons. The population of neurons is separated into a number of sub groups. Neurons are then selected from these separate groups and then combined to form the the complete neural network. During reproduction, neurons can only reproduce with other neurons within the same sub group. Offspring also remain within the parents' sub population. ESP also utilizes Delta-Coding which effectively acts as a local search to further refine the performance of the best found network. Whenever the sub-population diversity has reached a threshold diversity, delta-coding is applied so that new sub-populations are created based on the best found solution. This new set of new individuals are referred to as delta-chromosomes (Δ-chromosomes).

2.3 Neuro-Differential Evolution Algorithm

The Neuro Differential Evolution (NDE) algorithm is a more recent TWEANN algorithm (Mason et al. (2017)). NDE operates by addressing the optimisation of the neural network's topology and the networks synaptic weights individually. Here the network topology is evolved using a genetic algorithm (GA) while the networks synaptic weights are evolved using differential evolution (DE) (Storn and Price (1997)). As with NEAT, the number of hidden neurons does is not predefined by the user. Instead, the algorithm begins with a single hidden neuron and iteratively adds hidden neurons as needed. Networks of the same size are considered to be of the same species. New species with more hidden neurons emerge in two ways: 1) They are added at each iteration with a certain probability. 2) When none of the current networks increase in fitness. Adding neurons to the neural network in this way helps to keep the network size to a minimum. Only networks of the same species are capable of reproducing with one another. Similar to the NEAT algorithm, having multiple species of networks increases diversity.

3 Watershed Management

The Watershed management problem consists of a number of individuals, each of which seeks to maximize their consumption of the same finite resource (Yang et al. (2009)). Each individual seeks to take water from the same river for their own individual needs. This particular problem has many constraints relating to the amount of water than can be taken from the river by each individual. The problem also involves a dynamic environment as the amount of water available in the river is not fixed. The problem ultimately consists of maximizing the overall benefit of the water for everyone. There are a total of 6 agents that seek to maximize their consumption. The evolved network will have direct control over 4 of these consumers. The final 2 consumers are not directly controlled by the network and therefore reactive to the directly controlled consumers. The 4 directly controlled consumers are as follows: x_1 denotes the water taken from the river for municipal consumption by the city. Both x_4 and x_6 represent the water taken from the river for farming. x_2 denotes the water released for hydro power from the dam. These 4 variables will be the outputs of the evolved neural networks. The 2 consumers of water that are not controlled by the network are x_3 and x_5. These denote the water available to each surrounding ecosystem. The values of x_i must be chosen to maximize the benefit of the water to everyone. The utility of the water to each consumer is defined by the objective function in Equation 1.

$$f_i(x_i) = a_i x_i^2 + b_i x_i + c_i \tag{1}$$

Where a_i, b_i and c_i are constants corresponding to each consumer (i) and α_i denotes the lower bound of x_i in L^3 (Yang et al. (2009)). The indirect variables relating to the water available to the surrounding ecosystem, x_3 and x_5, are outlined in Equation 2.

$$x_3 = Q_2 - x_4 \tag{2a}$$
$$x_5 = x_2 + x_3 - x_6 \tag{2b}$$

Where Q_2 is the influx of water (L^3) into the tributary each month. The water consumed by each individual x_i is constrained by as follows:

$$\alpha_1 - x_1 \leq 0 \tag{3a}$$
$$\alpha_2 - Q_1 + x_1 \leq 0 \tag{3b}$$
$$x_2 - S - Q_1 + x_1 \leq 0 \tag{3c}$$
$$\alpha_4 - x_3 \leq 0 \tag{3d}$$
$$\alpha_3 - x_4 \leq 0 \tag{3e}$$
$$\alpha_4 - Q_2 + x_4 \leq 0 \tag{3f}$$
$$\alpha_6 - x_5 \leq 0 \tag{3g}$$
$$\alpha_5 - x_6 \leq 0 \tag{3h}$$
$$\alpha_6 - x_2 - x_3 + x_6 \leq 0 \tag{3i}$$

The parameter S denotes the volume of water the dam can store (L^3) and Q_1 denotes the influx of water into the main river per month (L^3).

This research will apply the neural network controller to 100 distinct states for Q_1, Q_2 and S found in (Mason et al. (2018)).

The outputs of the network must be scaled to a predefined range of values before it can be applied to the watershed problem. These ranges of the possible network outputs are outlined in Equation 4, from Equation 3.

$$\alpha_1 \leq x_1 \leq Q_1 - \alpha_2 \tag{4a}$$
$$\alpha_3 \leq x_4 \leq Q_2 - \alpha_4 \tag{4b}$$
$$0 \leq x_2 \leq S + Q_1 - \alpha_1 \tag{4c}$$
$$\alpha_5 \leq x_6 \leq S + Q_1 + Q_2 - \alpha_1 - \alpha_3 - \alpha_6 \tag{4d}$$

In order to evaluate the performance of the evolving network, the algorithm must have a fitness measure. By combining the objective functions for each individual (Equation 1) with a penalty function for any constraint violations, the function in Equation 5 is obtained that can provide a measure of performance.

$$F = \sum_{i=1}^{6} f_i(x_i) - \sum_{j=1}^{N} C(|h_j + 1|\delta_j) \tag{5}$$

In the above equation N corresponds to the constraints addressed with the penalty function in each training state, C is a constant $10E2$ (determined using parameter sweeps) that increases the impact of any constraint violation on the fitness of the network, h_i corresponds to the severity of each constraint violation, and finally δ indicates if there is a constraint violation. If a particular constraint is met $\delta = 0$, otherwise $\delta = 1$. The network inputs consist of the 3 flow rates. The network outputs 4 values corresponding to the water allocation for directly controlled consumers.

4 Results and Conclusion

All simulations were conducted for 10 statistical runs. A total of 10^6 networks were evaluated in per statistical run. The final fitness of each of the algorithms tested were compared using the t-test ($\alpha = 0.05$). The results reported were rounded to 4 places for readability. Each algorithm was able to find acceptable solutions without any constraint violations. As is illustrated in Figure 1, the NDE algorithm converges the fasted to the best solution. The next best performing algorithm is ESP followed by NEAT. The average final fitness of each algorithm is illustrated in Table 1. Figure 1 illustrates that NEAT has a faster initial convergence than ESP but then slows as the algorithm progresses in its operation. It also appears that the NEAT algorithm has not finished converging after 10^6 evaluations. In terms of consistency, NDE has a much lower standard deviation than ESP and NEAT, 2 and 3 orders of magnitude lower respectively. This indicates that NDE is consistent in its performance. NDE is statistically better when compared using statistical testing. To get a more complete view of the effectiveness of neuroevolution approaches against other approaches, the results presented here are also compared to a particle swarm optimisation (PSO) trained neural network from a previous study (Mason et al. (2018)). When compared to the PSO results, it is also found that NDE and ESP converge to a significantly higher fitness than the PS0 variant, (AR-PSOAWL). However AR-PSOAWL does significantly outperform NEAT in terms of the final fitness.

Table 1 Average fitness and standard deviation over 10 runs for the NDE, NEAT, ESP and PSO (Mason et al. (2018)) algorithms.

Algorithm	Avg ± StDev
NDE	24137.0401 ± 0.5242
NEAT	23712.3900 ± 188.8516
ESP	24087.3883 ± 58.8458
PSO (Mason et al. (2018))	23854.4598 ± 4.3446

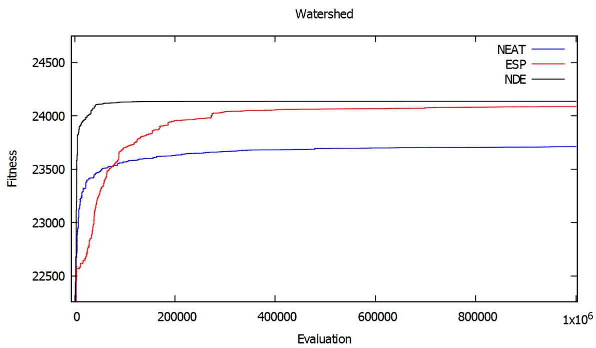

Fig. 1 Fitness convergence of the NEAT, ESP and NDE algorithms.

These results demonstrate that neuroevolution methods that explore different network configurations and sizes can lead to superior performance. There are many advantages when evolving neural network's connectivity and its synaptic weights rather than simply optimising the weights of the network, as previous research explores (Mason et al. (2018)). Firstly the network size does not need to be determined by the user when implementing NDE or NEAT. The algorithms are biased towards producing smaller networks. When simply applying PSO to the network weights, the user must determine the size of the network, usually by evaluating many network configurations. The second advantage of evolving the neural network's connectivity in addition to the synaptic weights, is that it allows the development of networks with fewer connections, which are easier to optimise and have fewer components for physical hardware implementations.

The following is now apparent as a result of this research:

1. Neuroevolution is an effective approach for watershed management, able to outperform the state of the art.
2. Optimising the neural network's connectivity along with its synaptic weights appears to provide better performance than solely optimising the network's synaptic weights for watershed management.
3. The NDE algorithm provides the best performance for the watershed management problem, out of the methods evaluated.

References

Barbalios N, Tzionas P (2014) A robust approach for multi-agent natural resource allocation based on stochastic optimization algorithms. Applied Soft Computing 18:12–24

Erfani T, Erfani R (2015) Fair resource allocation using multi-population evolutionary algorithm. In: European Conference on the Applications of Evolutionary Computation, Springer, pp 214–224

Floreano D, Dürr P, Mattiussi C (2008) Neuroevolution: from architectures to learning. Evolutionary Intelligence 1(1):47–62

Gomez FJ, Miikkulainen R (1999) Solving non-markovian control tasks with neuroevolution. In: IJCAI, vol 99, pp 1356–1361

Kasprzyk JR, Nataraj S, Reed PM, Lempert RJ (2013) Many objective robust decision making for complex environmental systems undergoing change. Environmental Modelling & Software 42:55–71

Mason K, Duggan J, Howley E (2017) Neural network topology and weight optimization through neuro differential evolution. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, ACM, pp 213–214

Mason K, Duggan J, Howley E (2018) A meta optimisation analysis of particle swarm optimisation velocity update equations for watershed management learning. Applied Soft Computing 62:148–161

Stanley KO, Miikkulainen R (2002) Evolving neural networks through augmenting topologies. Evolutionary computation 10(2):99–127

Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11(4):341–359

Yang YCE, Cai X, Stipanović DM (2009) A decentralized optimization algorithm for multiagent system–based watershed management. Water Resources Research 45(8)

View publication stats