Evolving Multi-Objective Neural Networks using Differential Evolution for Dynamic Economic Emission Dispatch

Karl Mason*
National University of Ireland Galway
Discipline of Information Technology
Galway, Ireland
k.mason2@nuigalway.ie

Jim Duggan
National University of Ireland Galway
Discipline of Information Technology
Galway, Ireland
jim.duggan@nuigalway.ie

Enda Howley
National University of Ireland Galway
Discipline of Information Technology
Galway, Ireland
ehowley@nuigalway.ie

ABSTRACT

This research presents a novel framework for evolving Multi-Objective Neural Networks using Differential Evolution (MONNDE). In recent years, the Differential Evolution algorithm has shown to be an effective and robust global optimisation algorithm. The algorithm uses evolutionary operators to optimise complex and continuous problem spaces and has been applied to a range of problems, recently including neural networks. This research continues this trend by utilizing differential evolution to evolve neural networks capable of addressing dynamic problems with multiple objectives. The proposed MONNDE framework is applied to the Dynamic Economic Emission Dispatch (DEED) problem. This problem consists of scheduling a group of power generators in a manner that minimises both cost and emissions produced by the generators. The power generators must also meet a series of constraints relating to their power output, power demand and network loss. The proposed MONNDE is performs very competitively when compared to algorithms such as NSGA-II, PSO, PSOAWL and MARL.

CCS CONCEPTS

•Mathematics of computing → Evolutionary algorithms; Bio-inspired optimization; **•Computing methodologies → Neural networks;** *Artificial intelligence; Genetic algorithms;*

KEYWORDS

Machine Learning, Dynamic economic dispatch, Dynamic economic emission dispatch, Multi-objective optimisation, Differential Evolution, Neural Networks

ACM Reference format:
Karl Mason, Jim Duggan, and Enda Howley. 2017. Evolving Multi-Objective Neural Networks using Differential Evolution for Dynamic Economic Emission Dispatch. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082480

*Corresponding Author

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3082480

1 INTRODUCTION

In recent years, Differential Evolution (DE) has shown to be an effective and robust global optimisation algorithm. It is a relatively new algorithm first proposed in 1997 by Storn and Price [32] and has had success in many areas since its first proposal, from energy [12] to robotics [7]. The algorithm uses evolutionary principles to iteratively search for the best solution to complex and continuous problems.

One such problem that differential evolution is well suited for is the training of Neural Networks (NN). Neural networks are biologically inspired function approximators that are routinely used in machine learning research [6]. They consist of layers of connected neurons that propagate an input signal through the network to produce some output. By training the weights of these networks, they can be used to approximate functions. This is useful for a wide range of problems, e.g. classification, control, forecasting, etc. [11]. This research will focus on applying differential evolution to train neural networks.

Much research has already been conducted applying differential evolution to both feed forward [1, 14] and recurrent neural networks [10, 26]. There has been no research however applying differential evolution to neural networks to address multi-objective problems. In standard single objective optimisation, optimisation algorithms are concerned with finding the single best solution that maximises or minimises some fitness function. Multi-objective optimisation is concerned with finding the range of solutions that optimise each objective with varying levels of significance. This solution set is known as the Pareto optimal set [16]. Multi-objective control is then concerned with dynamic problems where the environment is changing.

The problem that is the focus of this research is the Dynamic Economic Emission Dispatch (DEED) problem [5]. The DEED problem is a dynamic multi-objective scheduling problem in which power generators must be scheduled in a manner that both minimises operations cost and emission of harmful atmospheric pollutants. The task of power generation is significant for utility companies. It is vital that electricity is produced both in a cost-effective and environmentally friendly manner. It is also a very difficult task due to many different factors in the power generation process. These include: variation in the power demand throughout the day, power loss within the transmission lines, varying efficiency levels for different power generators with regard to cost and emissions, each generator has a different power generation limit and also a different ramp limit for increasing and decreasing its power output from hour to hour [5]. It is critically important that these power generators are scheduled efficiently due to the large potential increase in cost which would be incurred as a result of sup optimal power generator scheduling.

When generating electricity, the environmental cost must also be considered in addition to the financial cost. The emission of harmful atmospheric pollutants such as sulphur dioxide (SO_2) and nitrogen oxide (NO) is a familiar and heavily discussed problem in the world today [36]. Many countries world wide have pledged to reduce their negative impact on the environment [35]. Minimising the emissions of pollutants resulting from power generation can significantly contribute towards achieving this. The optimisation of both cost and emission from power stations is a conflicting problem however.As the optimisation algorithm approaches the optimal solution for each objective a trade off must be made. A solution that improves upon the cost of power generation will result in a deterioration in terms of the emissions produced and visa versa. Approaches to addressing these sorts of problems involve producing the Pareto Optimal set of solutions which are all considered equally optimal [16].

This research lies at the intersection between many different areas: evolutionary computing, multi-objective optimisation, machine learning and energy generation. The contributions of this paper are as follows:

(1) The proposal of a novel Multi-Objective Neural Network trained with Differential Evolution (MONNDE) framework. This framework will allow for the generation of the optimal set of solution for dynamic multi-objective problems.
(2) The application of the proposed MONNDE framework to the problem of power generation, in particular the DEED problem. The proposed MONNDE framework will be compared to current state of the art approaches.

The rest of this paper will be structured as followed: Section 2 will describe the Differential Evolution algorithm. Section 3 will give an overview on Neural Networks. Section 4 will outlined the area of Multi-Objective optimisation. Section 5 will detail the Dynamic Economic Emission Dispatch problem. Section 6 will describe the implementation of the proposed MONNDE framework The results of these experiments conducted will be presented in Section 7. Finally, Section 8 will draw conclusions based on these results and outline potential future research.

2 DIFFERENTIAL EVOLUTION

Differential Evolution is a state of the art global optimisation algorithm. The algorithm was first proposed by Storn and Price in 1997 [32]. DE was proposed for optimizing large and continuous problem spaces. An advantage of DE over more traditional gradient based methods is that DE does not rely on any gradient information and is applicable to noisy problems. The robustness and effectiveness of DE makes it a natural choice for optimizing network weights. DE has been applied to many real world problems such as robotics [7] and energy systems [12]. DE has previously been applied to neural network weight optimisation [1, 10, 14, 26] but not in a dynamic multi-objective setting. Differential evolution uses evolutionary methods to find the optimum solution.

At each iteration, the current position (solution) is combined with three other distinct positions to produce a new position y_i. This is described further in Algorithm 1. If the new position has a better fitness than the previous position, the previous position is replaced. This process is repeated for a predetermined number of iterations. Algorithm 1 outlines the algorithms operation where CR is the crossover probability and F is the differential weight.

```
Initialize X agents with random positions
while Iteration t <Tmax do
  for Agent = 1 to N do
    Select 3 other agents A,B and C
    Select random dimension index R
    for dimension = 1 to D do
      generate random number r ∈ [0,1]
      if r < CR Or i = R then
        new position y_i = a_i + F × (b_i − c_i)
      else
        y_i = x_i
      end if
    end
    if fitness(y) < fitness(x) then
      replace x with y
    end if
  end
end
Return best solution
```

Algorithm 1: Differential Evolution (DE) Algorithm

3 NEURAL NETWORKS

Neural Networks (NN), is a subfield of research within the field of Machine Learning and are inspired by the biological brain [6, 11]. Since their first proposal, NNs are have been applied to a range of problems such as classification, regression, control, online and offline learning and robotics. The standard feed forward network consists of an input layer of neurons, one or more hidden layer of neurons and an output layer. The network receives information in the form of a normalised signal into the input layer. This signal is carried through the connected layers of neurons via weighted synapses, or connections. The network then outputs the signal through the output layer. The most commonly used algorithm used to train the network weights is the backpropagation method [13]. However this method is only suitable for supervised learning as it requires a set of labelled data. This research will implement a particular type of neural network known as a Recurrent Neural Network (RNN), illustrated in Figure 1. Recurrent networks differ from the more standard feed forward networks in that they have recurrent connections between hidden neurons. These recurrent connections give the system memory which makes them particularly well suited to the DEED problem, which will be outlined later.

Information is fed into the network through the input layer of neurons. This signal is then passed to the neurons in the hidden layer of the network and is then outputted from the final output layer. This process is commonly referred to as a forward pass. At each forward pass, the hidden neurons retain information from the previous forward pass which is incorporated in their recurrent connections. By conducting parameter sweeps, it was found that a network configuration of 5 hidden neurons provided good performance for the DEED problem. This configuration consists of 80 network weights which must be optimised, i.e. 15 connections between the input and hidden layer, 20 recurrent connections and 45 connections between the hiddne and output layer.

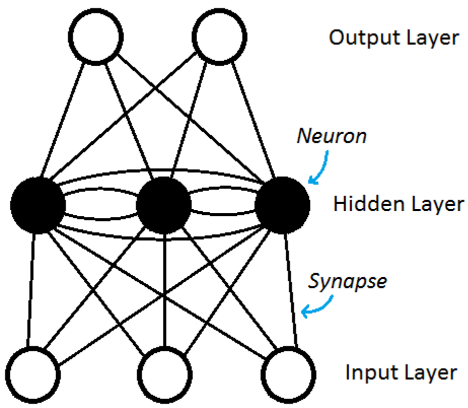

Figure 1: Recurrent Neural Network. This figure depicts the structure of a fully connected recurrent neural network. Neurons are connected by weighted connections (or synapses) that pass signals between neurons. The recurrent connections between the hidden neurons give the network memory.

As the signal is propagated through the network, its strength is adjusted by the weights between neurons. Aside from the input layer, a neuron in any other layer will have as input, the sum of the weighted signals that are outputted from other connected neurons. A neurons input is described in Equation 1.

$$v_j = \sum_{i=1}^{N} w_{i,j} a_i \tag{1}$$

Where v_j is the input to a neuron in the j^{th} layer, layer i is the preceding layer to j that contains N neurons, each neuron in layer i has output a_i and each of these output signals are weighted by the value $w_{i,j}$ as they are passed to each neuron in layer j.

Each neuron a_i outputs a value between 0 and 1. This output value is determined by the activation function of the neuron. The most commonly used action function is the sigmoid (or logistic) outlined in Equation 2

$$a_j = \frac{1}{1 + \exp -v_j} \tag{2}$$

4 MULTI-OBJECTIVE OPTIMISATION

Multi-objective optimisation is a subfield within optimisation research that is concerned with problems that contain two or more objectives. These problems have increased complexity due to the conflict that arrives when optimising these objectives. As the problem is optimised, there comes a point where by improving upon one object will result in a deterioration of the other objectives. Each of the solutions optimise the different objective with a varying level of significance. They are all considered equally optimal as long as they optimise at least one of the objectives better than any other solution. These optimal solutions are referred to as Pareto optimal solutions [16]. Figure 2 illustrates the Pareto optimal set.

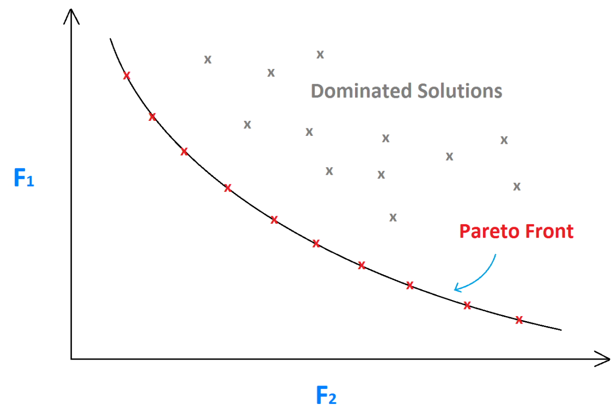

Figure 2: Pareto Optimal Front. This graph illustrates the location of the Pareto optimal front when two objectives F_1 and F_2 are being minimised. All of the solutions in grey are sub optimal and are dominated by the red Pareto optimal solutions.

4.1 Applications

The multi-objective framework has proven to be very popular in recent years. It has found a wide range of applications including: stock portfolio management [3], economics [33] and design [20]. The multi-objective paradigm presents the decision maker with a range of potential solutions rather than just a single optimal solution. The ability to present the decision maker with choice is advantageous, and even crucial, in many domains. A natural example would be for investment firms. Investing a clients finances is a balancing act between risk and potential profit. Multi-objective optimisation allows the investor to examine the expected returns if a high risk investment policy is implemented versus a safer investment policy.

4.2 Algorithms

Since the multi-objective paradigm came about, many of the well known single objective algorithms have been modified to fit the multi-objective framework. One of the most well known multi-objective algorithms is the Non-dominated Sorting Genetic Algorithm (NSGA-II) [8] which is an extension of commonly used genetic algorithms [4]. The Differential Evolution algorithm described in the previous section has been extended for multi-objective optimisation with the Pareto-frontier Differential Evolution (PDE) algorithm [2]. The well known Particle Swarm Optimisation algorithm [15] has been extended with the Multi Objective Particle Swarm Optimisation (MOPSO) variant [25]. There are numerous other examples such as Multi Objective Reinforcement Learning [34], Pareto Ant Colony Optimisation [9], Multi Objective Simulated Annealing [28], etc.

5 DYNAMIC ECONOMIC EMISSION DISPATCH

The DEED problem is a large and dynamic multi-objective optimisation problem and is therefore suitable to test the proposed MONNDE algorithm. The problem contains multiple constraints which makes the problem more difficult to solve. These include both hard and soft constraints. The problem also has equality constraints such as the power demand, and also inequality constraints such as the generator operation limits and ramp limits. The problem consists of optimising the scheduling of a group of power generators over a length of time in a manner that minimises both cost and emissions [5]. The cost function f_1 in Equation 3 represents the hourly running cost of all power generators.

$$f_1 = \sum_{i=1}^{N} [a_i + b_i P_{im} + c_i P_{im}^2 + |d_i sin\{e_i(P_i^{min} - P_{im})\}|] \quad (3)$$

Where $M = 24$ is the number of hours, $N = 10$ refers to the number of power generators, a_i, b_i, c_i, d_i and e_i are all cost coefficients associated with each generator i, the power output from generator i at time m is defined as P_{im} and the minimum possible power of generator i is defined as P_i^{min}. The emissions function f_2 in Equation 4 represents the amount of emissions produced by all power generators per hour.

$$f_2 = \sum_{i=1}^{N} [\alpha_i + \beta_i P_{im} + \gamma_i P_{im}^2 + \eta \exp \delta P_{im}] \quad (4)$$

Here α_i, β_i, γ_i, η_i and δ_i are the emission coefficients associated with each generator i. All solutions are subject to the equality constraint in Equation 5 that the total power output must be equal to the sum of the power demand and transmission loss.

$$\sum_{i=1}^{N} P_{im} = P_{Dm} + P_{Lm} \quad (5)$$

Where P_{Dm} represents the total power demand at time M and P_{Lm} represents the power loss within the transmission lines at time M. There are two inequality constraints which any potential solutions are subject to: generator operating limits and generator ramp limits. The operating limits are defined in Equation 6.

$$P_i^{min} \leq P_{im} \leq P_i^{max} \quad (6)$$

Here P_i^{max} and P_i^{min} refer to the maximum and minimum power output of each generator, $i \in N$ and $m \in M$. The ramp limits of each generator are outlined in Equation 8.

$$P_{im} - P_{i(m-1)} \leq UR_i \quad (7)$$
$$P_{i(m-1)} - P_{im} \leq DR_i \quad (8)$$

Here UR_i and DR_i are the ramp up and ramp down limits for each generator respectively, $i \in N$ and $m \in M$. The first power generator ($i = 1$) will be the slack generator, which will react to fluctuations in the power demand. The power output of the slack generator, P_{1m}, can be calculated by solving the quadratic equation 9.

$$\begin{aligned} 0 = B_{11} P_{1m}^2 + (2 \sum_{i=2}^{N} B_{1i} P_{im} - 1) P_{1m} + \\ (P_{Dm} + \sum_{i=2}^{N} \sum_{i=2}^{N} P_{im} B_{ij} P_{jm} - \sum_{i=2}^{N} P_{im}) \end{aligned} \quad (9)$$

Further details of the derivation of Equation 9 along with all coefficient values can be found in the work of M. Basu [5].

Each neural network output represents a power generator output at a given time. The slack power generator is not a network output as it is a reactive variable calculated using Equation 9. The power demand for the 24 hour period is illustrated in Equation 3.

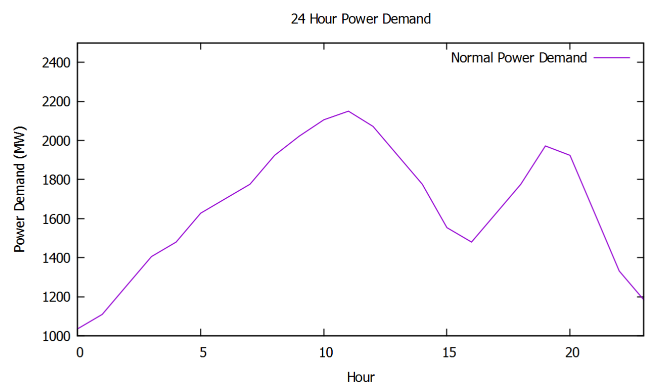

Figure 3: 24 Hour Power Demand. This graph illustrates the power demand that must be met by the power generators over a 24 hour period.

5.1 Constraint Handling

Balancing the generated power with the power demand and power loss is self constrained due to the slack power generator (Equation 5).

The power generator operating limits for the 9 non slack generators (Equation 6) will be handled by normalising the network outputs between the maximum and minimum possible generator outputs.

The slack generator operation limits constraint and generator ramp limit constraints will be enforced using the static penalty method [29] outlined in Equation 10. This penalty function will be incorporated into the objective function in order to train the network to avoid solutions which are not valid.

$$f_P = \sum_{i=1}^{N} C(|h_i + 1|\delta_i) \quad (10)$$

Were $N = 11$ is the number of constraints per hour handled using this method (1 slack generator operation limits and 10 generator ramp limits), $C = 10E6$ is the violation constant, h_i is the violation of each constraint and $\delta = 0$ if there is no constraint violation in a given dimension and $\delta = 1$ if a constraint is violated. The violation constant $C = 10E6$ was selected so that violations would have a significant impact on the fitness of the solution produced by the network.

6 MONNDE

This section will outline the proposed MONNDE framework (Multi-Objective Neural Network trained with Differential Evolution). The neural network implemented in this research is a fully connected recurrent neural network with 5 hidden neurons, as determined by parameter sweeps. The network will receive 3 inputs and have 9 outputs. The 3 inputs correspond to: 1) The current power demand. 2) The power demand at the previous time step. 3) The current value objective weight value w. The weight w will change from 0 to 1 in increments of 0.1. This will train the network to produce the Pareto optimal set of solutions that vary the importance of cost and emissions. The 9 outputs correspond to the 9 non slack power generators. The network will consist of a total of 80 connections, i.e. 80 weights that must be optimised by differential evolution.

In order for the deed problem to be optimised, the cost, emissions and penalty functions will be combined using a linear combination to form a single objective function to be minimised [27]. By varying the value of the objective weight w, the neural network will be trained to produce the Pareto optimal set.

$$F = wf_1 + (1-w)\lambda f_2 + f_p \tag{11}$$

The fitness function F is the hourly fitness function. Where f_1 is the cost function, f_2 is the emissions function, f_p is the penalty function, w is the weight and $\lambda = 10$ is the scaling factor [5]. The purpose of λ is to ensure each objective has equal influence.

The MONNDE algorithm will run for 10^6 iterations, to ensure that the network is fully trained. Each iteration will consist of evaluating 24 distinct states (power demands) as the objective weight w is varied from 0 to 1 for each state. A candidate position for the DE algorithm will be of size 80 and will correspond to a configuration of network weights. The fitness of a position corresponds to the cumulative fitness F (Equation 11) over the range of w over 24 hours. The crossover probability $CR = 0.9$ and differential weight $F = 0.5$, as determined by parameter sweeps.

7 RESULTS

In this section, the results of the conducted experiments will be presented. These will then be discussed and the performance of the proposed MONNDE will be compared to other state of the art approaches. The two tailed t-test was conducted to establish if the performance difference between any two algorithms is statistically significant. This is conducted using a significance level of 5%.

As depicted in Figure 4 shows the average 24 hour Pareto front. This figure clearly demonstrates that MONNDE is capable of successfully producing a range of solutions with varying levels of significance attributed to each objective. As previously mentioned, each solution along the Pareto front presented in Figure 4 is considered equally optimal.

Figures 5 & 6 illustrate the locations of the Pareto fronts produced by MMONDE on an hourly basis for hours 1 - 12 and 13 - 24 respectively. It can be seen how the locations of the Pareto optimal sets change hourly depending on the current demand for power. The large variance in power demand result in large variance in cost and emissions produced. The advantage of MONNDE over many of the previous methods in the literature, is its ability to produce

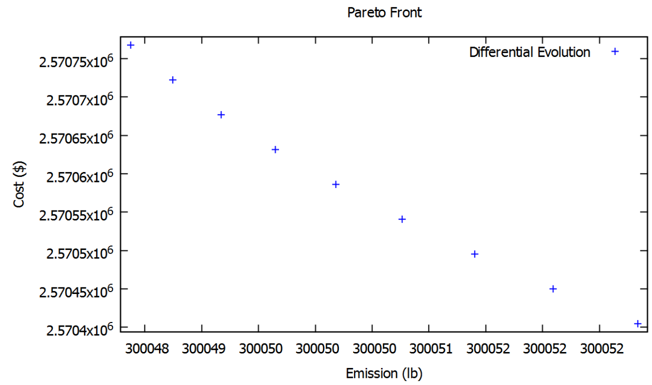

Figure 4: 24 Hour Pareto Front Averaged Over 10 Runs. This graphs illustrates the average Pareto front produced by the MONNDE over the total 24 hour period. Each point represents the total 24 hour cost and emissions for a particular weighting of each objective.

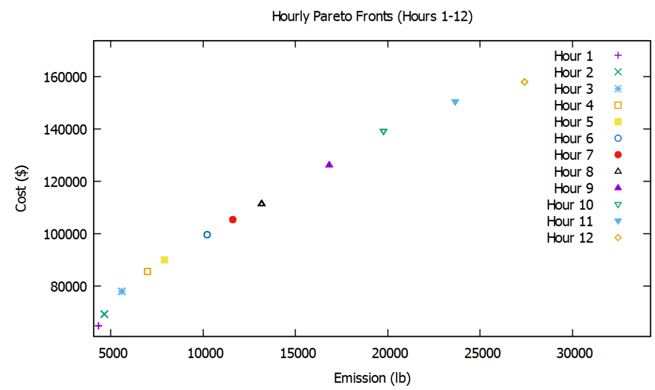

Figure 5: Locations of Pareto Fronts (Hours 1-12). This graph illustrates the locations of the Pareto optimal fronts for the cost and emissions for hours 1 - 12.

these hourly Pareto fronts as needed without any further optimisation needed. The proposed MONNDE approach approximates the problem as opposed to optimisation methods which simply find the optimum solution for the problem in its current form. This means that once MONNDE is trained to find the Pareto optimal solution of historic power demand information, it is capable of producing solutions as needed with no further training.

Table 1 presents the best solution produced by MONNDE when an equal weighting is given to cost and emissions ,i.e. $w = 0.5$ in Equation 11. In this table, the unit of power P is the megawatt (MW), cost is $\$ \times 10^6$ and emissions are $lb \times 10^5$. This table clearly demonstrates how MONNDE discovered that it is favorable to run particular power generators at maximum capacity constantly. Table 1 reveals that MONNDE found it to be optimal for power generators 7, 8, 9 & 10 to be maximising their outputs. This is consistent with previously observed results [5, 23].

Table 2 presents the best and average performance of MONNDE when objective weight $w = 0.5$. The average cost produced

Table 1: Best Solution for $w = 0.5$

Hour	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Cost	Emission	Violations
1	150.3707	135.0034	77.1484	90.3745	96.3520	121.5794	130.0000	120.0000	80.0000	55.0000	0.0626	0.0387	0
2	150.2386	135.0813	86.0664	114.6077	115.0397	146.4500	130.0000	120.0000	80.0000	55.0000	0.0659	0.0427	0
3	159.5925	140.5467	126.5353	164.3432	152.7878	157.8043	130.0000	120.0000	80.0000	55.0000	0.0742	0.0545	0
4	186.9033	169.7496	167.8469	195.5288	177.5522	159.5007	130.0000	120.0000	80.0000	55.0000	0.0842	0.0703	0
5	202.8397	190.0110	185.3834	208.4108	188.8184	159.8111	130.0000	120.0000	80.0000	55.0000	0.0895	0.0795	0
6	242.0817	226.1754	218.9330	234.0979	211.2370	159.9879	130.0000	120.0000	80.0000	55.0000	0.1005	0.1016	0
7	248.9272	246.7317	241.5329	250.5879	223.6012	159.9989	130.0000	120.0000	80.0000	55.0000	0.1063	0.1151	0
8	259.9971	268.0293	264.4474	265.5643	232.5495	159.9999	130.0000	120.0000	80.0000	55.0000	0.1127	0.1303	0
9	289.2804	322.5159	308.2810	288.6871	241.3566	160.0000	130.0000	120.0000	80.0000	55.0000	0.1257	0.1668	0
10	332.9179	360.7606	325.0965	295.4547	242.6199	160.0000	130.0000	120.0000	80.0000	55.0000	0.1381	0.1971	0
11	383.6473	391.3601	333.0023	298.1316	242.9070	160.0000	130.0000	120.0000	80.0000	55.0000	0.1493	0.2344	0
12	418.2559	402.7020	335.0032	298.7371	242.9498	160.0000	130.0000	120.0000	80.0000	55.0000	0.1568	0.2650	0
13	377.0143	368.3784	327.3842	296.2660	242.7218	160.0000	130.0000	120.0000	80.0000	55.0000	0.1452	0.2174	0
14	308.0193	313.6257	302.1461	285.8633	240.6315	160.0000	130.0000	120.0000	80.0000	55.0000	0.1262	0.1657	0
15	262.6231	268.3902	263.0722	264.5555	231.9882	159.9999	130.0000	120.0000	80.0000	55.0000	0.1129	0.1301	0
16	203.8970	216.4342	206.6278	224.1218	202.5693	159.9575	130.0000	120.0000	80.0000	55.0000	0.0942	0.0904	0
17	194.3953	192.9490	187.6793	210.0715	190.2678	159.8354	130.0000	120.0000	80.0000	55.0000	0.0892	0.0797	0
18	242.6655	226.1275	218.7349	233.9216	211.0837	159.9875	130.0000	120.0000	80.0000	55.0000	0.1006	0.1016	0
19	261.7306	267.1480	263.9773	265.3165	232.4293	159.9999	130.0000	120.0000	80.0000	55.0000	0.1128	0.1303	0
20	303.9873	343.9613	318.9194	293.1247	242.2643	160.0000	130.0000	120.0000	80.0000	55.0000	0.1312	0.1811	0
21	306.5244	314.2489	302.6769	286.1182	240.7018	160.0000	130.0000	120.0000	80.0000	55.0000	0.1262	0.1658	0
22	226.5929	234.3029	222.7162	236.1577	212.6291	159.9898	130.0000	120.0000	80.0000	55.0000	0.0998	0.1019	0
23	150.6551	157.4325	155.2731	186.4686	170.0162	159.1512	130.0000	120.0000	80.0000	55.0000	0.0778	0.0625	0
24	150.3991	136.1200	104.7717	142.3995	135.8462	154.8303	130.0000	120.0000	80.0000	55.0000	0.0698	0.0479	0
Cost	2.5518	Emission	2.9704										

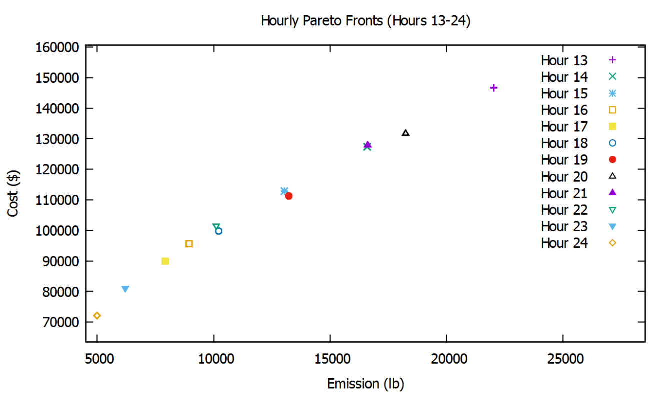

Figure 6: Locations of Pareto Fronts (Hours 13-24). This graph illustrates the locations of the Pareto optimal fronts for the cost and emissions for hours 13 - 24.

Table 2: NN-DE Average Cost and Emissions ($w = 0.5$)

Algorithm	Cost	Emissions
MONNDE (Best)	2.5518	2.9704
MONNDE (Avg)	2.5706	3.0005
SPSO [23]	2.6044	3.1075
PSOAWL [23]	2.5463	2.9455
NSGA-II [5]	2.5226	3.0994
MARL [18]	2.6641	3.3255

by MONNDE is 2.5706 ± 0.0146 while the average emissions is 3.0005 ± 0.0168. The performance of MONNDE is compared to both pure optimisation approaches (i.e., SPSO [15], PSOAWL [22, 24] and NSGA-II [8]) and a similar model based approach (i.e., MARL [19]). MONNDE is similar to MARL (Multi Agent Reinforcement Learning) in the sense that each approach involves developing a model of the dynamic optimisation problem. Each of these methods involve training a model to approximate the problem so that after a training period has been completed, the model can produce a solution for a given power demand with no further optimisation required. MONNDE performs significantly better than MARL in terms of both cost and emissions. Both the average and best solution produced by MONNDE dominates the best MARL solution.

When comparing the results of MONNDE to pure optimisation approaches, MONNDE performs better than SPSO, worse than PSOAWL and equal to NSGA-II. In multi-objective terms, the MONNDE solution dominates that of SPSO while being dominated by PSOAWL. MONNDE is non domimant when compared to NSGA-II, i.e., they are equally optimal. This demonstrates that the proposed MONNDE is highly competitive when compared to state of the art optimisation algorithms.

8 CONCLUSION

This research aimed to investigate developing a model capable of handling multiple objectives in a dynamic environment such as power generation. In particular, this research applied differential evolution to a recurrent neural network in order to approximate the dynamic economic emission dispatch problem. The proposed MONNDE framework (Multi-Objective Neural Network trained using Differential Evolution) is capable of producing the range of optimal solutions known as the Pareto optimal set. After training, the model developed by MONNDE is capable of producing a Pareto optimal solution set for any given power demand with no further optimisation required. This is not the case when simply applying optimisation algorithms. The proposed MONNDE performs significantly better than other model based approaches such as Multi Agent Reinforcement Learning (MARL) and produces highly competitive solutions when compared to state of the art optimisation algorithms such as Particle Swarm Optimisation (PSO) and Genetic Algorithms (GA). In summary, the contributions of this research are:

(1) A novel multi-objective neural network modelling framework is evolved using differential evolution. The proposed MONNDE has many potential applications such as sequential decision making, control and optimisation.
(2) The performance of the proposed MONNDE performs statistically better than other model based approaches such as MARL. When compared to solutions produced by state of the art optimisation algorithms (SPSO, PSOAWL and NSGA-II), MONNDE produces highly competative solutions.

8.1 Future Work

As a result from the research presented in this paper, there are many potential routes for future research. It is hoped than in future research, the proposed MONNDE framework can be applied to other multi-objective dynamic optimisation and control problems such as process optimisation, portfolio optimisation and production management.

Future research would also include evaluating the performance of the proposed multi-objective neural network framework with other optimisation algorithms for network weight optimisation, e.g., PSO [15] and Simulated Annealing [17].

It would be of interest in future work to evaluate the performance of other evolutionary approaches, in particular those that evolve the topology and weights of the network, e.g. NEAT [30, 31] and NDE [21]. By evolving the network topology in addition to the network weights, it may be possible to evolve networks with fewer weights capable of providing similar or increased performance.

REFERENCES

[1] Hussein A Abbass. 2002. An evolutionary artificial neural networks approach for breast cancer diagnosis. *Artificial intelligence in Medicine* 25, 3 (2002), 265–281.

[2] Hussein A Abbass, Ruhul Sarker, and Charles Newton. 2001. PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, Vol. 2. IEEE, 971–978.

[3] Fouad Ben Abdelaziz, Belaid Aouni, and Rimeh El Fayedh. 2007. Multi-objective stochastic programming for portfolio selection. *European Journal of Operational Research* 177, 3 (2007), 1811–1823.

[4] Thomas Bäck. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms.* Oxford university press.

[5] M Basu. 2008. Dynamic economic emission dispatch using nondominated sorting genetic algorithm-II. *International Journal of Electrical Power & Energy Systems* 30, 2 (2008), 140–149.

[6] Christopher M Bishop. 1995. *Neural networks for pattern recognition.* Oxford university press.

[7] Jayasree Chakraborty, Amit Konar, Lakhmi C Jain, and Uday K Chakraborty. 2009. Cooperative multi-robot path planning using differential evolution. *Journal of Intelligent & Fuzzy Systems* 20, 1, 2 (2009), 13–27.

[8] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International Conference on Parallel Problem Solving From Nature.* Springer, 849–858.

[9] Karl Doerner, Walter J Gutjahr, Richard F Hartl, Christine Strauss, and Christian Stummer. 2004. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of operations research* 131, 1-4 (2004), 79–99.

[10] Carlos A Duchanoy, Marco A Moreno-Armendáriz, Leopoldo Urbina, Carlos A Cruz-Villar, Hiram Calvo, and J de J Rubio. 2017. A Novel Recurrent Neural Network Soft Sensor via a Differential Evolution Training Algorithm for the Tire Contact Patch. *Neurocomputing* (2017).

[11] Simon S Haykin, Simon S Haykin, Simon S Haykin, and Simon S Haykin. 2009. *Neural networks and learning machines.* Vol. 3. Pearson Upper Saddle River, NJ, USA:.

[12] Dakuo He, Fuli Wang, and Zhizhong Mao. 2008. A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect. *International Journal of Electrical Power & Energy Systems* 30, 1 (2008), 31–38.

[13] Robert Hecht-Nielsen. 1989. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on.* IEEE, 593–605.

[14] Jarmo Ilonen, Joni-Kristian Kamarainen, and Jouni Lampinen. 2003. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters* 17, 1 (2003), 93–105.

[15] J. Kennedy and R. Eberhart. 1995. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Vol. 4. 1942–1948. DOI: http://dx.doi.org/10.1109/ICNN.1995.488968

[16] Il Yong Kim and OL De Weck. 2005. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization* 29, 2 (2005), 149–158.

[17] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, and others. 1983. Optimization by simmulated annealing. *science* 220, 4598 (1983), 671–680.

[18] Patrick Mannion, Karl Mason, Sam Devlin, Jim Duggan, and Enda Howley. 2016. Dynamic economic emissions dispatch optimisation using multi-agent reinforcement learning. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS 2016).*

[19] Patrick Mannion, Karl Mason, Sam Devlin, Jim Duggan, and Enda Howley. 2016. Multi-objective dynamic dispatch optimisation using multi-agent reinforcement learning. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, 1345–1346.

[20] R Timothy Marler and Jasbir S Arora. 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization* 26, 6 (2004), 369–395.

[21] Karl Mason, Jim Duggan, and Enda Howley. 2017. Neural Network Topology and Weight Optimization through Neuro Differential Evolution. In *Proceedings of the 2017 on Genetic and Evolutionary Computation Conference Companion.* ACM.

[22] Karl Mason and Enda Howley. 2015. Avoidance strategies in particle swarm optimisation. In *Mendel 2015.* Springer, 3–15.

[23] Karl Mason and Enda Howley. 2015. Avoidance techniques & neighbourhood topologies in particle swarm optimisation. *Master's thesis, National University of Ireland Galway* (2015).

[24] Karl Mason and Enda Howley. 2016. Exploring avoidance strategies and neighbourhood topologies in particle swarm optimisation. *International Journal of Swarm Intelligence* 2, 2-4 (2016), 188–207.

[25] Sanaz Mostaghim and Jürgen Teich. 2003. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE.* IEEE, 26–33.

[26] Sebastian Otte, Martin V Butz, Danil Koryakin, Fabian Becker, Marcus Liwicki, and Andreas Zell. 2016. Optimizing recurrent reservoirs with neuro-evolution. *Neurocomputing* 192 (2016), 128–138.

[27] Margarita Reyes-Sierra and CA Coello Coello. 2006. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research* 2, 3 (2006), 287–308.

[28] Paolo Serafini. 1994. Simulated annealing for multi objective optimization problems. In *Multiple criteria decision making.* Springer, 283–292.

[29] Alice E Smith, David W Coit, Thomas Baeck, David Fogel, and Zbigniew Michalewicz. 2000. Penalty functions. *Evolutionary computation* 2 (2000), 41–48.

[30] Kenneth O Stanley. 2002. Efficient reinforcement learning through evolving neural network topologies. In *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002.* Citeseer.

[31] Kenneth O Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary computation* 10, 2 (2002), 99–127.

[32] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.

[33] Ma Guadalupe Castillo Tapia and Carlos A Coello Coello. 2007. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on.* IEEE, 532–539.

[34] Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. 2011. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine learning* 84, 1-2 (2011), 51–80.

[35] Vigdis Vestreng, Gunnar Myhre, Hilde Fagerli, Stefan Reis, and Leonor Tarrasón. 2007. Twenty-five years of continuous sulphur dioxide emission reduction in

Europe. *Atmospheric chemistry and physics* 7, 13 (2007), 3663–3681.

[36] Suthathip Yaisawarng and J Douglass Klein. 1994. The effects of sulfur dioxide controls on productivity change in the US electric power industry. *The Review of Economics and Statistics* 76, 3 (1994), 447–460.