

# Optimizing carbon tax for decentralized electricity markets using an agent-based model

Anonymized

## ABSTRACT

Averting the effects of anthropogenic climate change requires a transition from fossil-fuels to low-carbon technology. A way to achieve this is to decarbonise the electricity grid. This would reduce carbon emissions due to electricity generation, and also help to decarbonise other sources such as automotive and heating by enabling a low-carbon alternative. Carbon taxes have been shown to be an efficient way to aid in this transition.

In this paper we demonstrate how to find optimal carbon tax policies by using a genetic algorithm approach to model the electricity market agent-based model ElecSim. To achieve this we use the NSGA-II genetic algorithm to minimise average electricity price and relative carbon intensity of the electricity mix. We demonstrate that it is possible to find a range of carbon taxes to suit different objectives. We minimise electricity cost to below £10/MWh and reduce carbon intensity to 0 in all cases. The optimal carbon tax strategies generally increase from 2020 to 2035, and are above £81/tCO<sub>2</sub> in all instances, with a mean of £240/tCO<sub>2</sub>.

## KEYWORDS

Energy markets, policy, carbon tax, genetic algorithm, optimization, digital twin, agent-based models, electricity market model

### ACM Reference Format:

Anonymized. 2020. Optimizing carbon tax for decentralized electricity markets using an agent-based model. In *Tenth ACM International Conference on Future Energy Systems*, June 22–26, 2020, Melbourne, Australia. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3185768.3186313>

## 1 INTRODUCTION

Computer simulation allows practitioners to model real-world systems using software. These simulations allow for ‘what-if’ analyses which can provide an indication as to how a system may behave under certain policies, environments and assumptions. These simulations become particularly important in systems which have high costs, impacts or risks associated with them. A ‘digital twin’ is a concept which has emerged in recent years. This is defined as a simulation of a specific instance of a system.

Electricity markets are an example of such a system. Disruptions to electricity supply, a substantial increase in the cost of electricity or unrestrained carbon emissions have the potential to

destabilise economies [13, 19]. It is for reasons such as these that electricity market models are used to test hypotheses, develop strategies and gain an understanding of underlying dynamics to prevent undesirable consequences [12].

In this paper we use the electricity market agent-based model ElecSim to find an optimum carbon tax policy [14]. Specifically, we use a genetic algorithm to find a carbon tax policy to reduce both average electricity price and the relative carbon density by 2035 for the UK electricity market.

Carbon taxes have been shown to quickly lower emissions and lower the costs to the public [24]. Carbon taxes are able to send clear price signals, as opposed to a cap-and-trade scheme, such as the EU Emissions Trading System, which has shown to be unstable [24].

In this paper we use the reference scenario projected by the UK Government’s Department for Business & Industrial Strategy (BEIS) with model parameters calibrated by Kell *et al.* [8, 15]. This reference scenario projects energy and emissions until 2035. We undertook various carbon tax policy interventions to see how we could reduce relative carbon density whilst at the same time reduce the average electricity price.

The parameter space we optimise over is the carbon tax price over a 17 year period from 2018 to 2035. The carbon price is used to influence the objectives of average electricity price and relative carbon intensity in 2035. Grid and random search are approaches which trial parameters at evenly distributed spaces and random spaces respectively. These approaches, however, are often inefficient, however, require an increased number of simulations due to their non-dynamic nature. Genetic Algorithms in contrast use an evolutionary computing approach to find global optimal solutions faster.

In order to optimise over two potentially competing objectives, i.e. average electricity price and relative carbon intensity, we use the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [6]. The NSGA-II algorithm can approximate a Pareto frontier [22, 23]. A Pareto frontier is a curve in which there is no solution which is better than another along the curve for different sets of parameters. In this context, better means that a solution is closer to the optimal for a particular combination of objectives.

We find that the rewards of average electricity price and relative carbon intensity are not mutually exclusive. That is, it is possible to have both a lower average electricity price and a lower relative carbon price. This is due to the low short-run marginal cost of renewable technology, which has been shown to lower electricity prices [21].

The main contribution of this paper is to explore carbon tax strategies using genetic algorithms for multi-objective optimisation. We optimise both average electricity price and relative carbon intensity of the electricity market over a 17 year timeframe for the UK.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*e-Energy* '20, June 22–26, 2020, Melbourne, Australia

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/3185768.3186313>

The following sections are set out as follows. Section 2 covers examples of optimisations using genetic algorithms and different carbon strategies. Section 3 details the optimization techniques applied. Section 4 explores the electricity market model used. We present our results in Section 5, and conclude in Section 6.

## 2 LITERATURE REVIEW

Multi-objective optimisation problems are commonplace. In this section we review multiple applications that have used multi-objective optimisation, as well as explore the literature which focus on finding optimal carbon tax strategies.

### 2.1 Examples of Optimization

Similarly to our work, Ascione *et al.* use the NSGA-II algorithm to generate a Pareto front to optimise for two objectives: operating cost for space conditioning and thermal comfort [2]. The aim of this paper is to optimise the hourly set point temperatures with a day-ahead planning horizon. A Pareto front is generated which allows a user to select a solution according to their comfort needs and economic constraints. This work showed a reduction in operating costs by up to 56% as well as improved thermal comfort.

Gorzałczany *et al.* also apply the NSGA-II algorithm, however, they apply it to the credit classification problem [10]. The objectives optimised over were accuracy and interpretability when making financial decisions such as credit scoring and bankruptcy prediction. This technique was able to significantly outperform the alternative methods in terms of interpretability while remaining competitive or superior in terms of the accuracy and speed of decision making in comparison with the existing classification methods.

### 2.2 Carbon Tax Strategies

In this section we explore different strategies employed in the literature to analyse the benefits and consequences of a carbon tax. However, we were unable to find a study which employed a multi-objective optimisation algorithm to minimise average electricity price and relative carbon density.

Levin *et al.* use an optimisation model to analyse market and investment impacts of several incentive mechanisms to support investment in renewable energy and carbon emission reductions [17]. Carbon tax was found to be the most cost-efficient method of reducing emissions.

Zhou *et al.* construct a social welfare model based on a Stackelberg game [25]. The differences and similarities between a flat carbon tax and an increasing block tariff carbon tax are analysed using a numerical simulation. This work shows that an increasing block tariff carbon tax policy can significantly reduce tax burdens for manufacturers and encourage low-carbon production. In contrast to Zhou *et al.* we trial multiple different carbon tax strategies using a machine learning approach.

Li *et al.* use a hierarchical carbon market scheduling model to reduce carbon emissions [18]. Multi-objective optimisation was applied to discover optimal behaviours for policy makers, customers and generators to minimise the costs incurred by these actors. Our work, however, focuses on the different strategies of carbon tax as opposed to optimal actor behaviour.

---

#### Algorithm 1 Genetic algorithm [3]

---

```

1:  $t = 0$ 
2: initialize  $P_t$ 
3: evaluate structures in  $P_t$ 
4: while termination condition not satisfied do
5:    $t = t + 1$ 
6:   select reproduction  $C_t$  from  $P_{t-1}$ 
7:   recombine and mutate structures in  $C_t$ 
     forming  $C'_t$ 
8:   evaluate structures in  $C'_t$ 
9:   select each individual for  $P_t$  from  $C'_t$ 
     or  $P_{t-1}$ 
10: end while

```

---

## 3 OPTIMIZATION METHODS

Multi-objective optimisation allows practitioners to overcome the problems with optimising multiple objectives with classical optimisation techniques. Multi-objective optimisation algorithms are able to generate Pareto-optimal solutions as opposed to converting the multiple objectives into a single-objective problem. A Pareto frontier is made up of many Pareto-optimal solutions which can be displayed graphically. A user is then able to choose between various solutions and trade-offs according to their wishes.

The NSGA-II algorithm, a multi-objective genetic optimisation algorithm, is able to generate a Pareto frontier in a single optimisation run.

In the following sub-sections we detail the standard genetic algorithm followed by the NSGA-II algorithm.

### 3.1 Genetic Algorithms

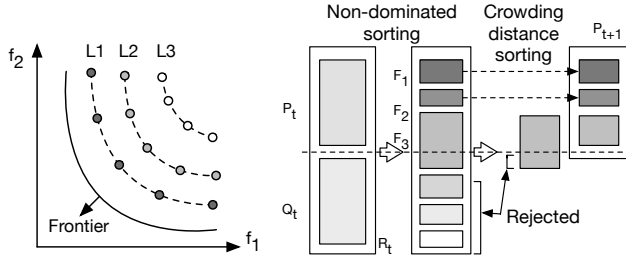
Genetic Algorithms (GAs) [11] are a class of evolutionary algorithm which can be used for optimisation. In this sub-section we detail the generic version of a genetic algorithm.

As shown by Algorithm 1, initially, a population of structures  $P_0$  is generated.  $P_0$  is then evaluated for fitness. Where fitness in this case is the reward that is to be optimised. Next, a subset of individuals from  $P_0$  are chosen for mating,  $C_{t+1} \subset P_t$ . This subset of individuals are selected proportional to their fitness. For mating with the subset  $C_{t+1}$ , the 'fitter' individuals have a higher chance of reproducing to create the offspring group  $C'_{t+1}$ . The individuals of  $C'_{t+1}$  have characteristics dependent on the genetic operators, crossover and mutation [20]. These genetic operators are an implementation decision [3]. The new population  $P_{t+1}$  is then created by merging individuals from  $C'_{t+1}$  and  $P_t$ . See Algorithm 1 for detailed pseudocode.

### 3.2 NSGA-II

NSGA-II is efficient for multi-objective optimization on a number of benchmark problems and finds a better spread of solutions than Pareto Archived Evolution Strategy (PAES) [16] and Strength Pareto EA (SPEA) [26] when approximating the true Pareto-optimal front [6].

The majority of multi-objective optimisation algorithms use the concept of *domination* during population selection [5]. A



**Figure 1: a) Schematic of non-dominated sorting with solution layering b) Schematic of the NSGA-II procedure**

non-dominated algorithm, however, seeks to achieve the Pareto-optimal solution. This is where no single solution should dominate another. An individual solution  $\mathbf{x}^1$  is said to dominate another  $\mathbf{x}^2$ , if and only if there is no objective of  $\mathbf{x}^1$  that is worse than objective of  $\mathbf{x}^2$  and at least one objective of  $\mathbf{x}^1$  is better than the same objective of  $\mathbf{x}^2$  [4].

Non-domination sorting is the process of finding a set of solutions which do not dominate each other and make up the Pareto front. See Figure 1a for a visual representation, where  $f_1$  and  $f_2$  are two objectives to minimise and L1, L2 and L3 are dominated layers.

In this section we define the processes used by the NSGA-II algorithm to determine which solutions to keep:

**3.2.1 Non-dominated sorting.** We assume that there are  $M$  objective functions to minimise, and that  $\mathbf{x}^1 = \{x_j^1\}$  and  $\mathbf{x}^2$  are two solutions.  $x_j^1 < x_j^2$  implies solution  $\mathbf{x}^1$  is better than solution  $\mathbf{x}^2$  on objective  $j$ . A solution  $\mathbf{x}^1$  is said to dominate the solution  $\mathbf{x}^2$  if the following conditions are true:

- (1) The solution  $\mathbf{x}^1$  is no worse than  $\mathbf{x}^2$  in every objective. I.e.  $x_j^1 \leq x_j^2 \quad \forall j \in \{1, 2, \dots, M\}$ .
- (2) The solution  $\mathbf{x}^1$  is better than  $\mathbf{x}^2$  in at least one objective. I.e.  $\exists j \in \{1, 2, \dots, M\} \text{ s.t. } x_j^1 < x_j^2$ .

Next, each of the solutions are ranked according to their level of non-domination. An example of this ranking is shown in Figure 1a. Here,  $f_1$  and  $f_2$  are the objectives to be minimised. The Pareto front is the first front. All of the solutions in the Pareto front are not dominated by any other solution. The solutions in layer 1, L1, are dominated only by those in the Pareto front, and are non-dominated by those in L2 and L3.

The solutions are then ranked according to their layer. For example, the solutions in the Pareto front are given a fitness rank ( $i_{rank}$ ) of 1, solutions in L1 have an  $i_{rank}$  of 2.

**3.2.2 Density Estimation.** ( $i_{distance}$ ) is calculated for each solution. This is the average distance between the two closest points to the solution in question.

**3.2.3 Crowded comparison operator.** ( $<_n$ ) is used to ensure that the final frontier is an evenly spread out Pareto-optimal front. This is achieved by using the two attributes: ( $i_{rank}$ ) and ( $i_{distance}$ ). The partial order is then defined as:

$i <_n j$  if ( $i_{rank} < j_{rank}$ ) or (( $i_{rank} = j_{rank}$ ) and ( $i_{distance} > j_{distance}$ )) [6].

This order prefers solutions with a lower rank  $i_{rank}$ . For solutions with the same rank, the solution in the less dense area is preferred.

**3.2.4 Main loop.** Similarly to the standard GA, a population  $P_0$  is created with random parameters. The solutions of  $P_0$  are then sorted according to non-domination. The child population  $C'_1$  of size  $N$  is then created by binary tournament selection, recombination and mutation operators. In this case, tournament selection is the process of evaluating and comparing the fitnesses of various individuals within a population. In binary tournament selection, two individuals are chosen at random, the fitnesses are evaluated and the individual with the better solution is selected [1].

Next, a new combined population is formed  $R_t = P_t \cup C'_t$ .  $R_t$  has a size of  $2N$ .  $R_t$  is then sorted according to non-domination. A new population is then formed  $P_{t+1}$ . Solutions are added from the sorted  $R_t$  in order of non-domination. Solutions are added until the size of  $P_{t+1}$  exceeds  $N$ . The solutions from the last layer are prioritised based on having the largest crowding distance [6].

This process is shown in Figure 1b, which is repeated until the termination condition is met. A termination condition could be: no significant improvement over  $X$  iterations or a specified number of iterations have been performed. The full procedure is detailed formally by Algorithm 2.

## 4 EXPERIMENTAL SETUP

### 4.1 Simulation Environment

In order to evaluate the different carbon strategies we used the model developed by Kell *et al.*, ElecSim [14, 15]. ElecSim is an agent-based model which mimics the behaviour of decentralised electricity markets. For this paper, we have parametrised the model to data for the UK in 2018 to act as a digital twin of the UK electricity market. This includes the power plants in operation in 2018, and the funds available to their respective companies [7, 9].

Six fundamental sections make up ElecSim: 1) power plant data; 2) scenario data; 3) the time-steps of the algorithm; 4) the power exchange; 5) the investment algorithm and 6) the generation companies (GenCos) as agents. ElecSim uses a subset of representative days of electricity demand, solar irradiance and wind speed to

---

#### Algorithm 2 NSGA-II main loop [6]

---

- 1:  $R_t = P_t \cup C'_t$  combine parent and child population
  - 2:  $\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$   
where  $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$
  - 3:  $P_{t+1} = \emptyset$
  - 4: **while** do  $|P_{t+1}| < N$
  - 5:   Calculate the crowding distance of ( $\mathcal{F}_i$ )
  - 6:    $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$
  - 7: **end while**
  - 8: Sort( $P_{t+1}, <_n$ ) sort in descending order using  $<_n$
  - 9:  $P_{t+1} = P_{t+1}[0 : N]$  select the first  $N$  elements of  $P_{t+1}$
  - 10:  $Q_{t+1} = \text{make-new-population}(P_{t+1})$  using  
selection, crossover and mutation to create  
the new population  $Q_{t+1}$
  - 11:  $t = t + 1$
-

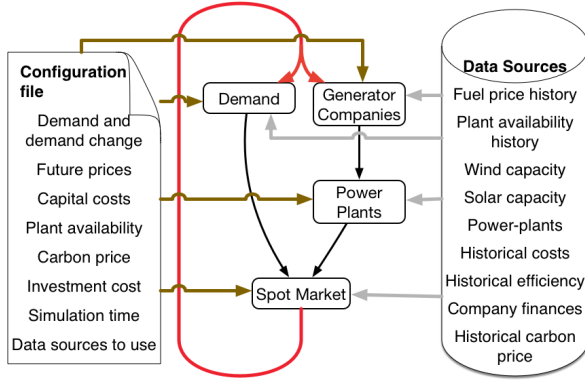


Figure 2: System overview of ElecSim.

approximate a full year. In this context, representative days are a subset of days which when scaled up can adequately represent a year. Figure 2 details how these components interact.

Specifically, the configuration file details the scenario which can be set by the user. This includes electricity demand, carbon price and fuel prices. The data sources parametrise the digital twin to a particular country, including information such as wind capacity and power plants in operation. Generation Companies own and invest in power plants. These power plants are then matched to electricity demand using a spot market.

The market runs in merit-order dispatch and bids are made by the power plant's short-run marginal cost (SRMC). Investment in power plants are based upon a net present value (NPV) calculation. NPV is able to evaluate and compare investments with cash flows spread over many years. This is shown formally in Equation 1, where  $t$  is the year of the cash flow,  $i$  is the discount rate,  $N$  is the total number of years, or lifetime of power plant, and  $R_t$  is the net cash flow of the year  $t$ :

$$NPV(i, N) = \sum_{t=0}^N \frac{R_t}{(1+i)^t}. \quad (1)$$

The yearly income for each power plant is estimated for each generation company by running a merit-order dispatch electricity market 10 years into the future. However, the expected cost of electricity 10 years into the future is uncertain. We therefore use the reference scenario projected by BEIS and use the predicted costs of electricity calibrated by Kell *et al* [8, 15]. The agents predict the future carbon price by using a linear regression model.

## 4.2 Optimization

**4.2.1 Non-parametric carbon policy.** The optimization approach has two stages. First, we initialize the population of the NSGA-II algorithm  $P_0$  with 18 attributes. These correspond to a separate carbon tax for each year, shown by Equation 2:

$$P_0 = \{a_1, a_2, \dots, a_{18}\}, 0 \leq a_y \leq 250, \quad (2)$$

where  $P_0$  is the first population,  $a_y$  is the attribute or carbon price in year  $y$  and  $a_1$  is the carbon price in year 1,  $a_2$  the carbon price in year 2 and so forth. Each of the carbon prices are bound between the values of £0 and £250. This provides the optimization

algorithm with the highest degree of freedom. This high degree of freedom enables a high number of strategies to be trialled due its non-parametric nature. This, however, comes with a large search space requiring a large number of iterations.

**4.2.2 Linear carbon policy.** To reduce the search space for the carbon strategy, we trial a linear carbon strategy, of the form:

$$p_c = a_1 y_t + a_2, -14 \leq a_1 \leq 14, 0 \leq a_2 \leq 250, \quad (3)$$

where  $p_c$  is the carbon price,  $y_t$  is the year,  $a_1$  is the gradient or first attribute and  $a_2$  is the intercept or second attribute.  $a_1$  is bound by  $-14$  and  $14$ , and  $a_2$  by  $0$  and  $250$ . These bounds are chosen to ensure that the carbon price does not exceed  $\sim£500$  in the year 18 (2035) and is greater than about  $-\£250$ , as well as ensuring that the carbon tax in the first year is greater than £0 but smaller than £250.

## 5 RESULTS

In this section we explore the results of the optimisations, the optimum carbon strategies and the resultant electricity mixes.

### 5.1 Non-parametric carbon policy

Figure 3 displays the development of the genetic algorithm against the rewards, relative carbon density and average electricity price. Darker colours display higher generation numbers. The first generation shows a wide spread in relative carbon density and average electricity price. However, over successive generations the solutions converge to a relative carbon density of 0 and an average electricity price under £10MW/h.

Strikingly, the rewards of relative carbon density and average electricity price are not mutually destructive. This could be due to the low short-run marginal cost of renewable energy which reduces both electricity prices and carbon emissions [21].

To understand the optimum carbon strategies we visualised the parameters that produced the lowest average electricity prices in Figure 4. Specifically, we filtered for electricity prices under £5/MWh and displayed the results using a heat map. The darker colours represent a higher density of points.

Figure 4 displays a general trend, where carbon tax starts at  $\sim£100$  until the year 2030, where it increases to  $\sim£200$  by 2035. This may be due to the fact that a lower initial carbon tax of  $\sim£100$  encourages investment in low-carbon technologies before the higher rate of  $\sim£200$  comes into force. This higher rate of carbon tax would allow GenCos to outcompete higher carbon emitting generators over the lifetime of the plants.

### 5.2 Linear carbon policy

Figure 5 displays the development of the genetic algorithm against the rewards: relative carbon density and average electricity price. Similarly to the non-parametric carbon policy shown in Figure 3, the first generation shows a wide spread of results. However, the spread is smaller than that of the linear carbon policy. This may be due to the fact that it is easier for the GenCos to predict the carbon policy, which increases confidence in the NPV calculations. The linear carbon policy also converges to a relative carbon density of 0, and an average electricity price smaller than £10MW/h.

Figure 6 compares the distributions of average electricity price for both techniques. Both methods show improvements

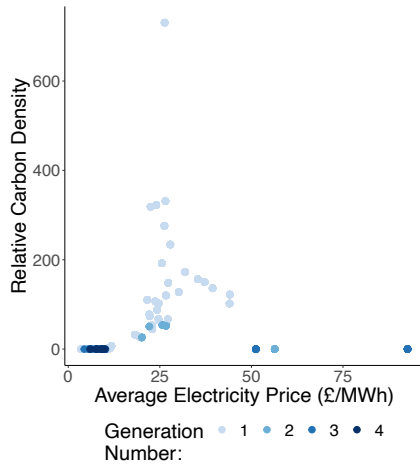


Figure 3: Development of genetic algorithm rewards for non-parametric carbon tax policy results in 2035..

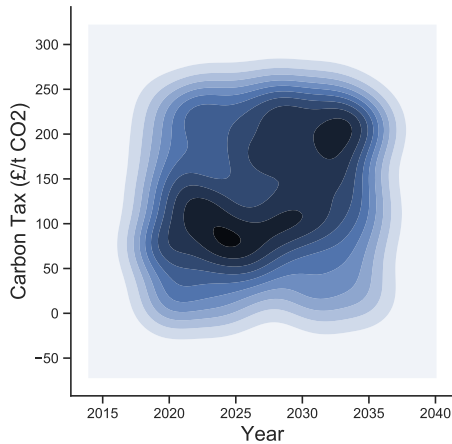


Figure 4: Density plot of points with an average electricity price <£5/MWh for non-parametric carbon tax policy results in 2035.

as the number of generations of the genetic algorithm increase. The linear policy, however, is able to more quickly converge to a low average electricity price, with a mode of ~£5.4MW/h. The non-parametric policy has a number of poorer performing parameters, and Generation Number 4 has a bimodal distribution, with a mode of ~£6.3MW/h.

Figure 7a displays the linear carbon policies which had an average electricity price under £4.5MW/h. There is no single 'optimum' carbon policy; a range of policies are able to achieve low carbon and a low average electricity price.

We explore the electricity mix generated of three different strategies shown in Figure 7b. We selected the highest, lowest and the lowest flat carbon strategy to show a range of possible strategies.

Figure 8 displays the generated electricity mixes for each of the selected strategies. To generate these images we ran 80 scenarios to capture the variability between scenarios.

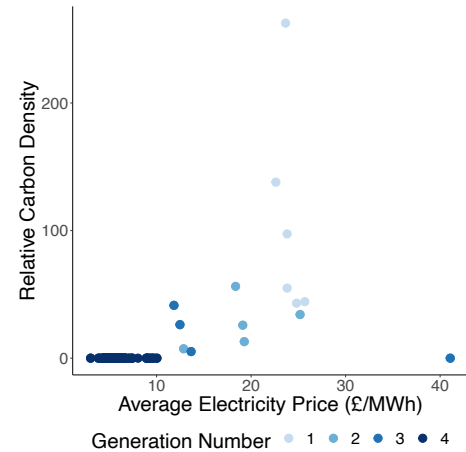


Figure 5: Development of genetic algorithm rewards in 2035 for linear carbon strategy.

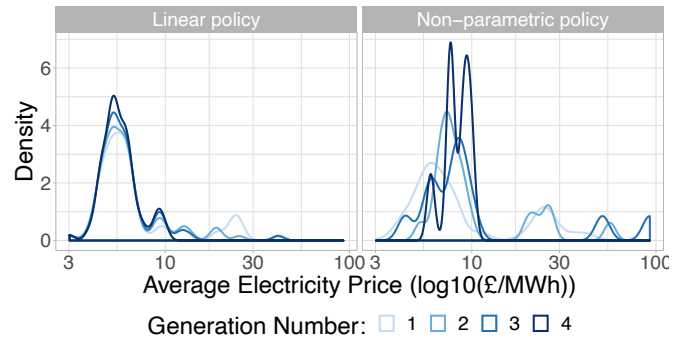


Figure 6: Density plot of average electricity price in 2035 over generation number of genetic algorithm for both linear and non-parametric policy.

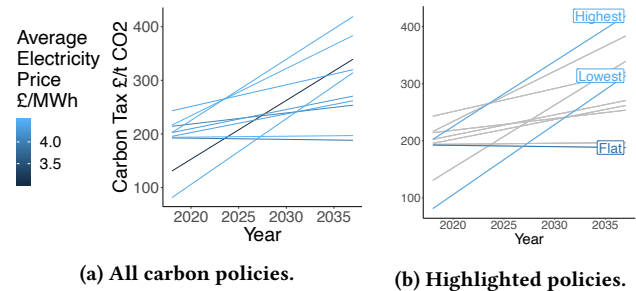
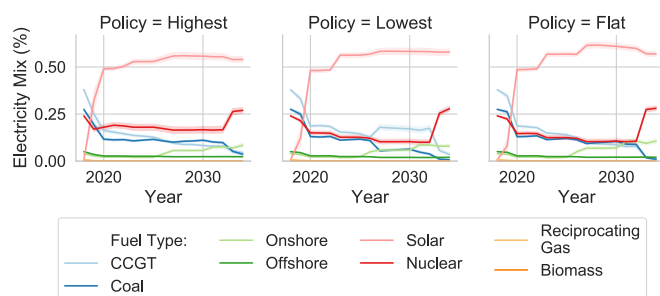


Figure 7: Linear carbon policies under £4.5MW/h visualised.

Whilst there does not seem to be a significant difference between scenarios, with solar providing ~60% of electricity mix by 2035, there is an observable difference with the other generator types.



**Figure 8: Electricity mixes under selected linear carbon policies.**

The ‘highest’ carbon strategy exhibits a higher uptake in nuclear, possibly due to the fact that nuclear becomes more competitive when compared to coal or gas. The ‘lowest’ carbon strategy shows a higher uptake in Combined Cycle Gas Turbines (CCGT) during the years of 2026 to 2031 as it outcompetes nuclear. The ‘flat’ carbon policy shows a higher percentage of solar energy than any of the other scenarios, albeit with a lower percentage of nuclear. Onshore wind is shown to be consistent for these scenarios.

## 6 CONCLUSION

In this paper we have demonstrated that it is possible to use the genetic algorithm technique NSGA-II to optimise carbon tax policy using an electricity market agent-based model.

We trialled a non-parametric carbon policy by allowing the genetic algorithm to optimise a carbon price for each year. These results showed us that a linear carbon tax may be appropriate. We then used a linear model as a carbon tax policy to reduce the total number of parameters for the genetic algorithm to optimise.

We were able to show that a range of linear carbon taxes were able to achieve both low average electricity price and a relative carbon intensity of 0 in 2035. By exploring three different carbon tax policies we saw that ~60% of electricity consumption in the UK would be provided by solar. The difference between these ‘optimal’ carbon tax policies was largely shown by competition between CCGT, coal and nuclear.

This was largely due to the low short-run marginal cost of solar and nuclear energy which means that they are often dispatched ahead of the fossil-fuel based generators. CCGT and coal, however, are useful for filling demand when there is low solar irradiance.

In future work we would like to try additional scenarios with varying future generation costs and calculate a sensitivity analysis to carbon taxes.

## REFERENCES

- [1] Rosshairy Abd Rahman, Razamin Ramli, Zainoddin Jamari, and Ku Ruhana Ku-Mahamud. 2016. Evolutionary Algorithm with Roulette-Tournament Selection for Solving Aquaculture Diet Formulation. *Mathematical Problems in Engineering* 2016 (2016), 1–10. <https://doi.org/10.1155/2016/3672758>
- [2] Fabrizio Ascone, Nicola Bianco, Claudio De Stasio, Gerardo Maria Mauro, and Giuseppe Peter Vanoli. 2016. Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort. *Energy and Buildings* 111 (2016), 131–144. <https://doi.org/10.1016/j.enbuild.2015.11.033>
- [3] Thomas Back, David B Fogel, and Zbigniew Michalewicz. 2009. Evolutionary Computation 1 Basic Algorithms and Operators. *Comprehensive Chemometrics* (2009). <https://doi.org/10.1016/B978-044452701-1.09002-5>
- [4] Chunten Bao, Lihong Xu, Erik D. Goodman, and Leilei Cao. 2017. A novel non-dominated sorting algorithm for evolutionary multi-objective optimization. *Journal of Computational Science* 23 (2017), 31–43. <https://doi.org/10.1016/j.jocs.2017.09.015>
- [5] Edmund K. Burke and Kendall Graham. 2014. *Search methodologies: Introductory tutorials in optimization and decision support techniques, second edition*. 1–716 pages. <https://doi.org/10.1007/978-1-4614-6940-7>
- [6] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T Meyarivan. 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II. *CEUR Workshop Proceedings* 1133 (2000), 850–857.
- [7] Energy & Industrial Strategy Department for Business. 2019. Companies House - GOV.UK. *UK Government* (2019).
- [8] Department for Business Energy & Industrial Strategy. 2019. Updated energy and emissions projections 2018. *The Energy White Paper* April (2019).
- [9] Department for Business Energy and Industrial Strategy. 2019. Power stations in the United Kingdom, May 2019. *Digest of United Kingdom Energy Statistics (DUKES)* (2019).
- [10] Marian B. Gorzalczyk and Filip Rudziński. 2016. A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability. *Applied Soft Computing Journal* 40 (2016), 206–220. <https://doi.org/10.1016/j.asoc.2015.11.037>
- [11] John H. Holland. 1975. *Search methodologies: Introductory tutorials in optimization and decision support techniques, 2nd ed.*
- [12] S. Jebaraj and S. Iniyan. 2006. A review of energy models. *Renewable and Sustainable Energy Reviews* 10, 4 (2006), 281–311. <https://doi.org/10.1016/j.rser.2004.09.004>
- [13] Nyasha Kaseke and Stephen G Hosking. 2013. Sub-Saharan Africa Electricity Supply Inadequacy: Implications. 29, 2 (2013), 113–132. <https://doi.org/10.1353/eas.2013.0009>
- [14] Alexander Kell, Matthew Forshaw, and A Stephen McGough. 2019. ElecSim : Monte-Carlo Open-Source Agent-Based Model to Inform Policy for Long-Term Electricity Planning. *The Tenth ACM International Conference on Future Energy Systems (ACM e-Energy)* (2019), 556–565.
- [15] Alexander J. M. Kell, Matthew Forshaw, and A. Stephen McGough. 2020. Long-Term Electricity Market Agent Based Model Validation using Genetic Algorithm based Optimization (Under Submission). (2020).
- [16] Joshua Knowles and David Corne. 1999. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999* 1 (1999), 98–105. <https://doi.org/10.1109/CEC.1999.781913>
- [17] Todd Levin, Jonghwan Kwon, and Audun Botterud. 2019. The long-term impacts of carbon and variable renewable energy policies on electricity markets. *Energy Policy* 131, February (2019), 53–71. <https://doi.org/10.1016/j.enpol.2019.02.070>
- [18] Dan Li, Weiqi Hua, Hongjian Sun, and Wei Yu Chiu. 2017. Multiobjective optimization for carbon market scheduling based on behavior learning. *Energy Procedia* 142 (2017), 2089–2094. <https://doi.org/10.1016/j.egypro.2017.12.581>
- [19] V Masson-Delmotte, P Zhai, H.O Pörtner, D Roberts, J Skea, P R Shukla, A Pirani, W Moufouma-Okia, C Péan, R Pidcock, S Connors, J B Matthews, Y Chen, X Zhou, M I Gomis, E Lonnoy, T Maycock, M Tignor, and T Waterfield. 2018. *IPCC Special Report 1.5 - Summary for Policymakers*. IPCC.
- [20] Melanie Mitchell. 1998. *An introduction to genetic algorithms*. MIT press.
- [21] Amy O’Mahoney and Eleanor Denny. 2011. The Merit Order Effect of Wind Generation on the Irish Electricity Market. *Usae 2011* 56043 (2011), 1–12. <https://doi.org/10.1227/01.NEU.0000349921.14519.2A> arXiv:suresh govindarajan
- [22] Vilfredo Pareto and Ann S (Translator) Schiwer. 1927. *Manual of political economy Tr. by Ann S. Schiwer*. Macmillan, London, London and Basingstoke.
- [23] W. Stadler. 1979. A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960. *Journal of Optimization Theory and Applications* 29, 1 (1979), 1–52. <https://doi.org/10.1007/BF00932634>
- [24] Bettina B.F. Wittebren. 2009. Exon is right: Let us re-examine our choice for a cap-and-trade system over a carbon tax. *Energy Policy* 37, 6 (2009), 2462–2464. <https://doi.org/10.1016/j.enpol.2009.01.029>
- [25] Dequn Zhou, Yunfei An, Donglan Zha, Fei Wu, and Qunwei Wang. 2019. Would an increasing block carbon tax be better? A comparative study within the Stackelberg Game framework. *Journal of Environmental Management* 235, July 2018 (2019), 328–341. <https://doi.org/10.1016/j.jenvman.2019.01.082>
- [26] Eckart Zitzler and Lothar Thiele. 2006. Multiobjective optimization using evolutionary algorithms - A comparative case study. (2006), 292–301. <https://doi.org/10.1007/bfb0056872>