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

Anonymized

ABSTRACT

Placeholder

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 *ICPE '20: ACM/SPEC International Conference on Performance Engineering, April 20–24, 2020, Edmonton, Canada.* ACM, New York, NY, USA, 5 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.

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. 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 [**?**].

In this paper we use the electricity market agent-based model ElecSim to find an optimum carbon tax strategy [**?**]. 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 lower emissions quickly with lower costs to the public, with no upper bounds in terms of the reduction potential [**?**]. 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.

ICPE '20, April 20–24, 2020, Edmonton, Canada

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/3185768.3186313

In this paper we use the reference scenario projected by the UK Government's Department for Business & Industrial Strategy (BEIS) and used the model parameters calibrated by Kell *et al.* [**? ?**]. 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.

In contrast to grid or random search, genetic algorithms have the ability to converge on an optimal solution by trialling a fewer number of parameters [**?**]. Grid search is the trialling of parameters at evenly distributed spaces and random search is where parameters are chosen at random. This is of particular importance in cases with a large number of parameters or in simulations which require a long compute time which is the case for ElecSim.

In order to optimise over these two potentially competing objectives, we used a Genetic Algorithm, the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [**?**]. The NSGA-II algorithm can approximate a Pareto frontier [**? ?**]. 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.

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 [**?**].

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.

The rest of this paper is set out as follows. Section 2, Section 3, Section 4, Section 5, and 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

Ascione *et al*. use the NSGA-II algorithm to generate a Pareto front to optimise operating cost for space conditioning and thermal comfort [**?**]. They aim to optimise the hourly set point temperatures with a day-ahead planning horizon. They generate a Pareto front which allows a user to select a solution according to their comfort needs and economic constraints.

They are able to reduce operating costs by up to 56% as well as improve thermal comfort.

Gorzałczany *et al*. apply the NSGA-II algorithm to the credit classification problem [**?**]. They optimise over accuracy and interpretability when making financial decisions such as credit scoring and bankruptcy prediction. They are 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.

Ma *et al*. use the multi-objective artificial immune optimization algorithm for land use allocation (MOAIM-LUA model) [**?**]. They balance land use supply and demand based on the future dynamic demands from different land use stakeholders in the region at the macro-level in Anlu County, China. They optimise the objectives of economic benefits and ecological benefits. They found that for this application they were able to obtain better solution sets than the NSGA-II algorithm.

2.2 Carbon Tax Strategies

Zhou *et al*. construct a social welfare model based on a Stackelberg game [**?**]. They analyse the differences and similarities between a flat carbon tax and an increasing block tariff carbon tax using a numerical simulation. They show 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 [**?**]. They use multi-objective optimisation to find optimal behaviours for policy makers, customers and generators to minimise the costs incurred by these actors. Our paper, however, focuses on the different strategies of carbon tax as opposed to optimal actor behaviour.

Levin *et al*. use an optimisation model to analyze market and investment impacts of several incentive mechanisms to support investment in renewable energy and carbon emission reductions [**?**]. They find that a carbon tax is the most cost-efficient method of reducing emissions.

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 this section we will first detail the standard genetic algorithm, followed by a look at NSGA-II.

Algorithm 1 Genetic algorithm [**?**]

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**

3.1 Genetic Algorithms

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

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. These genetic operators are an implementation decision [**?**].

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) [**?**] and Strength Pareto EA (SPEA) [**?**] when approximating the true Pareto-optimal front [**?**].

The majority of multi-objective optimisation algorithms use the concept of *domination* during population selection [**?**]. A 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 $\mathbf{x}^2$ [**?**].

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:

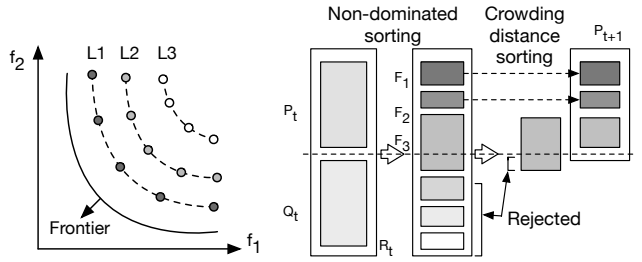

Figure 1: a) Schematic of non-dominated sorting with solution layering b) Schematic of the NSGA-II procedure

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. $x_j^1 \leq x_j^2 \;\; \forall j \in \{1, 2, \ldots, M\}$.

(2) The solution $\mathbf{x}^1$ is better than $\mathbf{x}^2$ in at least one objective. I.e. $\exists\, j \in \{1, 2, \ldots, M\} \;\; s.t. \;\; x_j^1 < x_j^2$.

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, etc.

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. ($\prec_n$) 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:
$i \prec_n j$ if ($i_{rank} < j_{rank}$) or (($i_{rank} = j_{rank}$) and ($i_{distance} > j_{distance}$)) [**?**].

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 [**?**].

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 [**?**].

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.

4 EXPERIMENTAL SETUP

4.1 Simulation Environmental

In order to evaluate the different carbon strategies we used the model developed by Kell *et al.*, ElecSim [**? ?**]. 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. This includes the power plants in operation in 2018, and the funds available to their respective companies [**? ?**].

Five 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 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 parts interact.

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:

Algorithm 2 NSGA-II main loop [**?**]

$R_t = P_t \cup C_t'$ combine parent and child population
$\mathcal{F}$ = fast-non-dominated-sort (R_t)
    where $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$
$P_{t+1} = \emptyset$
while do $|P_{t+1} < N|$
    Calculate the crowding distance of ($\mathcal{F}_i$))
    $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$
end while
Sort($P_{t+1}, \prec_n$) sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1}[0 : N]$ select the first N elements of P_{t+1}
Q_{t+1} = make-new-population(P_{t+1}) using
    selection, crossover and mutation to create
    the new population Q_{t+1}
$t = t + 1$

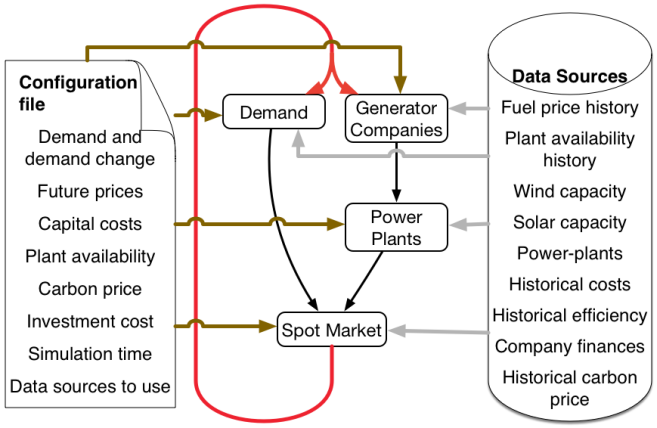

Figure 2: System overview of ElecSim.

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

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* [**? ?**].

4.2 Optimization

4.2.1 Non-parametric carbon strategy. The optimization approach took two stages. Firstly, we initialized the population of the NSGA-II algorithm P_0 with 18 attributes. These corresponded to a separate carbon tax for each year, shown by Equation 2:

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

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 were bound between the values of £0 and £250 . This provided the optimization algorithm with the highest degree of freedom. This high degree of freedom enabled a high number of strategies to be trialled due its non-parametric nature. This, however, came with a large search space requiring a large number of iterations.

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

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

Figure 3: Development of genetic algorithm rewards of average electricity price and relative carbon density in 2035 over time for highest degrees of freedom per year.

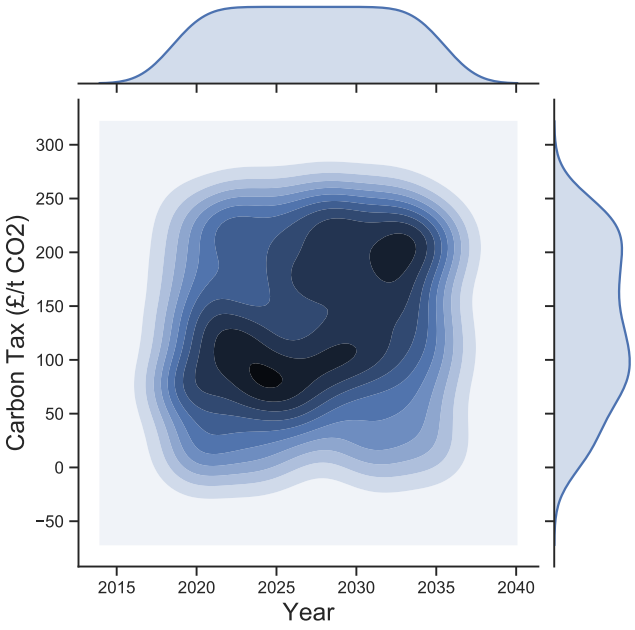

Figure 4: 2D density plot of carbon tax strategies that led to an average electricity price of below £5/MWh by 2035.

where p_c is the carbon price, y_t is the year, m is the gradient or first attribute and c is the intercept or second attribute. a_1 was bound by −14 and 14, and a_2 by 0 and 250. These bounds were chosen to ensure that the carbon price did not exceed ~£500 in the year 18 (2035) and was greater than ~ − £250, as well as ensuring that the carbon tax in the first year was greater than 0 but smaller than 250.

5 RESULTS

6 CONCLUSION

Placeholder

ACKNOWLEDGMENTS

Anonymized

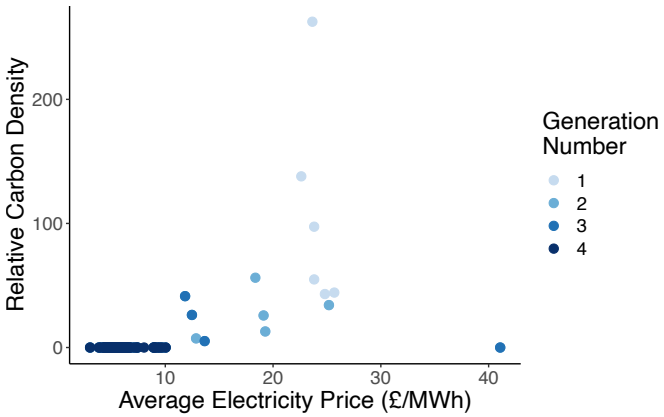

Figure 5: Development of genetic algorithm rewards of average electricity price and relative carbon density in 2035 over time for linear carbon strategy.

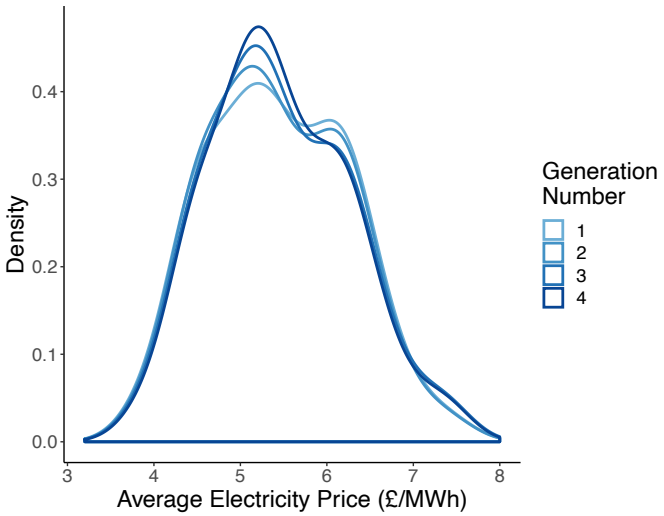

Figure 6: Density plot of average electricity price smaller than £8/MWh in 2035 over generation number of genetic algorithm.

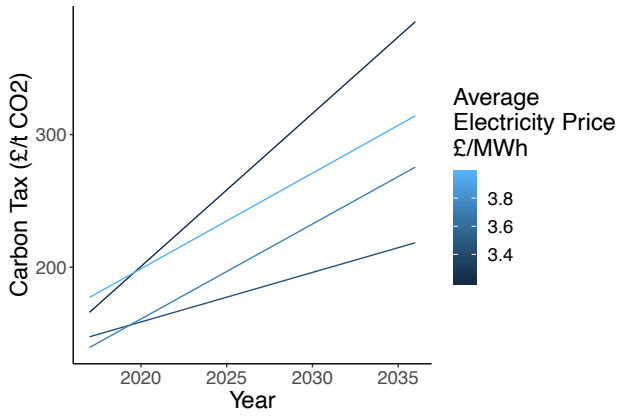

Figure 7: Linear carbon tax strategies visualised with average electricity price smaller than £5/MWh.