

# 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 [11].

In this paper we use the electricity market agent-based model ElecSim to find an optimum carbon tax strategy [12]. 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 [21]. 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.

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.* [8, 13]. 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 [5]. 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) [7]. The NSGA-II algorithm can approximate a Pareto frontier [19, 20]. 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 [18].

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 [2]. 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 [9]. 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.

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).

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

© 2020 Association for Computing Machinery.

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

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

Ma *et al.* use the multi-objective artificial immune optimization algorithm for land use allocation (MOAIM-LUA model) [17]. 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 [22]. 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 [16]. 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 [15]. 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.

### 3.1 Genetic Algorithms

Genetic Algorithms (GAs) [10] 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

---

#### 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

```

---

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 [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) [14] and Strength Pareto EA (SPEA) [23] when approximating the true Pareto-optimal front [7].

The majority of multi-objective optimisation algorithms use the concept of *domination* during population selection [6]. 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  $\mathbf{x}^2$  [4].

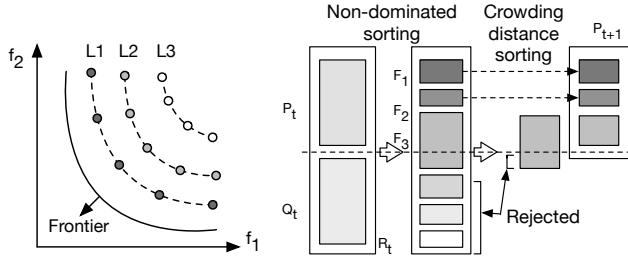
Non-domination sorting is the process of finding a set of solutions which do not dominate each other and make up the Pareto front. See Figure 1a for a visual representation, where  $f_1$  and  $f_2$  are two objectives to minimise and L1, L2 and L3 are dominated layers.

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  $\mathbf{x}^1 = \mathbf{x}^1_j$  and  $\mathbf{x}^2$  are two solutions.  $x^1_j < x^2_j$  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.  $x^1_j \leq x^2_j \quad \forall j \in \{1, 2, \dots, M\}$ .
- (2) The solution  $\mathbf{x}^1$  is better than  $\mathbf{x}^2$  in at least one objective. I.e.  $\exists j \in \{1, 2, \dots, M\} \text{ s.t. } x^1_j < x^2_j$ .

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



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

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.** ( $<_n$ ) 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 ( $i_{distance} > j_{distance}$ )) [7].

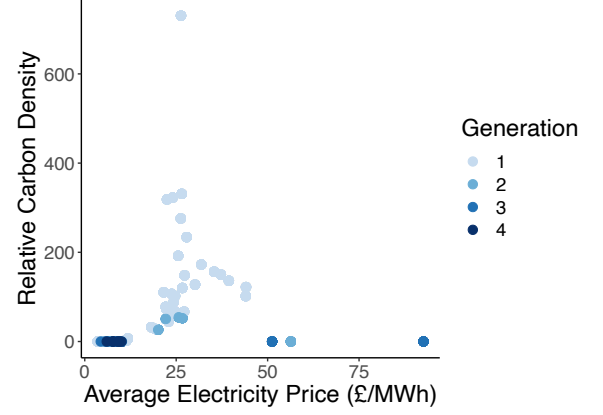
This order prefers solutions with a lower rank  $i_{rank}$ . For solutions with the same rank, the solution in the less dense area is preferred.

**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

After the first population the procedure changes (see Algorithm 2). Initially, a combined population is formed  $R_t = P_t \cup C'_t$  of size  $2N$ .  $R_t$  is sorted according to non-domination. A new population is now formed ( $P_{t+1}$ ), adding solutions from each front level until the size of  $P_{t+1}$  exceeds  $N$ . The solutions of the last accepted level are then sorted according to  $<_n$ , and a total of  $N$  solutions are chosen, rejecting those from the last layer that have a smaller crowding distance [7].

The entire process is shown in Figure 1b, and is repeated until the termination condition is met. Termination conditions could be: no significant improvement over  $X$  iterations or a specified number of iterations have been performed.



**Figure 2: Development of genetic algorithm rewards of average electricity price and relative carbon density in 2035 over time for highest degrees of freedom per year.**

## 4 SIMULATION ENVIRONMENT

Placeholder

## 5 RESULTS

## 6 CONCLUSION

Placeholder

## ACKNOWLEDGMENTS

Anonymized

## 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*

---

### Algorithm 2 NSGA-II main loop [7]

---

- 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:  $t = t + 1$
-

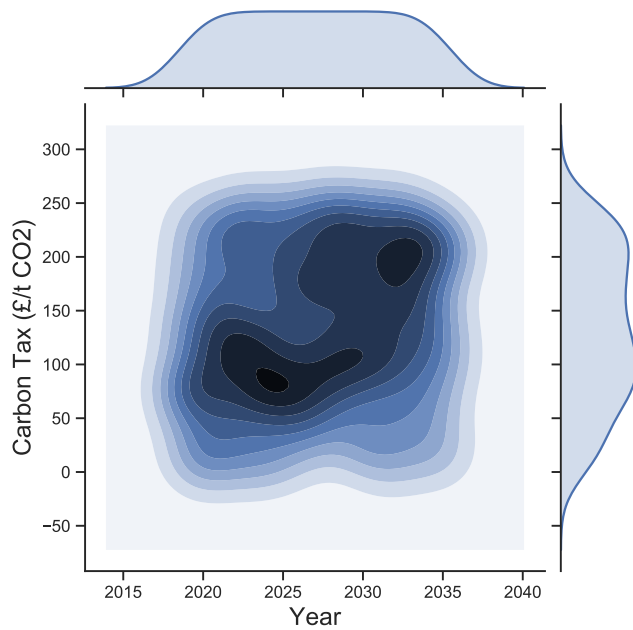


Figure 3: 2D density plot of carbon tax strategies that led to an average electricity price of below £5/MWh by 2035.

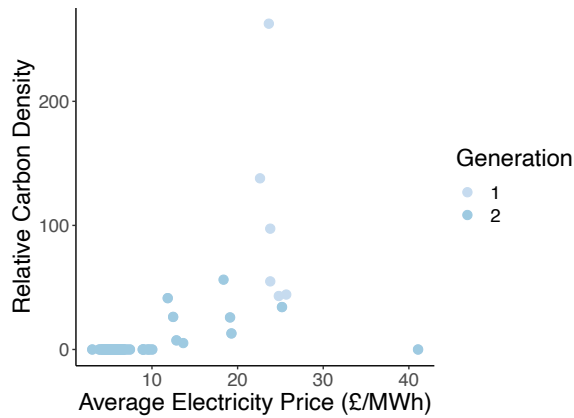


Figure 4: Development of genetic algorithm rewards of average electricity price and relative carbon density in 2035 over time for linear carbon strategy.

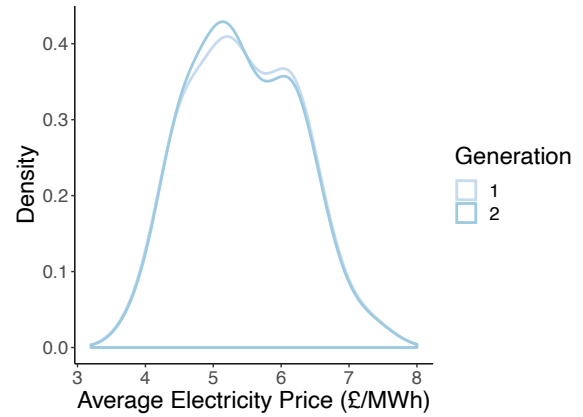


Figure 5: Density plot of average electricity price smaller than £8/MWh in 2035 over generation number of genetic algorithm.

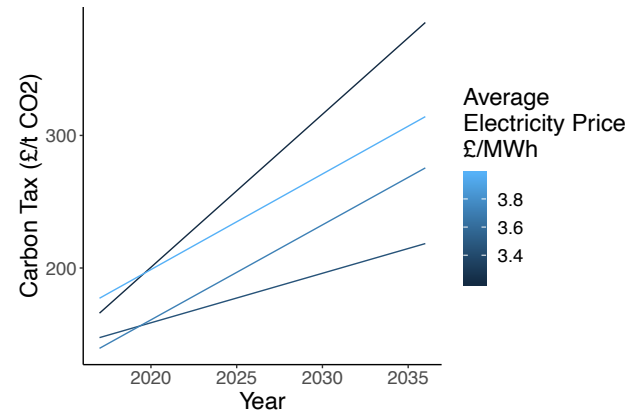


Figure 6: Linear carbon tax strategies visualised with average electricity price smaller than £5/MWh.

- (2009). <https://doi.org/10.1016/B978-044452701-1.09002-5>
- [4] Chunteng 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] James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-Parameter Optimization. 13 (2012), 1–25. [papers3://publication/uuid/1190E1AB-0319-40C5-81CD-7207784965DE](https://arxiv.org/abs/1106.0960)
  - [6] 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>
  - [7] 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.
  - [8] Department for Business Energy & Industrial Strategy. 2019. Updated energy and emissions projections 2018. *The Energy White Paper* April (2019).
  - [9] 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>
  - [10] John H. Holland. 1975. *Search methodologies: Introductory tutorials in optimization and decision support techniques, 2nd ed.*
  - [11] 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>
  - [12] 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.
  - [13] 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).
  - [14] 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>
  - [15] 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>

- [16] 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>
- [17] Xiaoya Ma and Xiang Zhao. 2015. Land use allocation based on a multi-objective artificial immune optimization model: An application in Anlu county, China. *Sustainability (Switzerland)* 7, 11 (2015), 15632–15651. <https://doi.org/10.3390/su71115632>
- [18] 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
- [19] Vilfredo Pareto and Ann S (Translator) Schwier. 1927. *Manual of political economy Tr. by Ann S. Schwier*. Macmillan, London, London and Basingstoke.
- [20] 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>
- [21] Bettina B.F. Wittneben. 2009. Exxon 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>
- [22] 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>
- [23] Eckart Zitzler and Lothar Thiele. 2006. Multiobjective optimization using evolutionary algorithms – A comparative case study. (2006), 292–301. <https://doi.org/10.1007/bfb0056872>