ETH zürich

Modelling the impact of carbon tax policies across multiple economic regions

Wen Yi Chan, Joel Huber, Gaspard Krief, Tianyi Liu, Pascal Troxler

19 December 2022 | Complex Social Systems: Modeling Agents, Learning, and Games

Press release of the EU:

CBAM (Carbon Border Adjustment Mechanism) targets imports of products in carbon-intensive industries. The objective of CBAM is to prevent that the greenhouse gas emissions reduction efforts of the EU are offset by increasing emissions outside its borders through relocation of production to non-EU countries (where policies applied to fight climate change are less ambitious than those of the EU) or increased imports of carbon-intensive products.

https://www.consilium.europa.eu/en/press/press-releases/2022/12/13/eu-climate-action-provisional-agreement-reached-on-carbon-border-adjustment-mechanism-cbam/, 13/12/2022

Related work

Regulation at the source? Comparing upstream and downstream climate policies

Joël Foramitti [a,b,*], Ivan Savin [a,c], Jeroen C.J.M. van den Bergh [a,b,d,e]

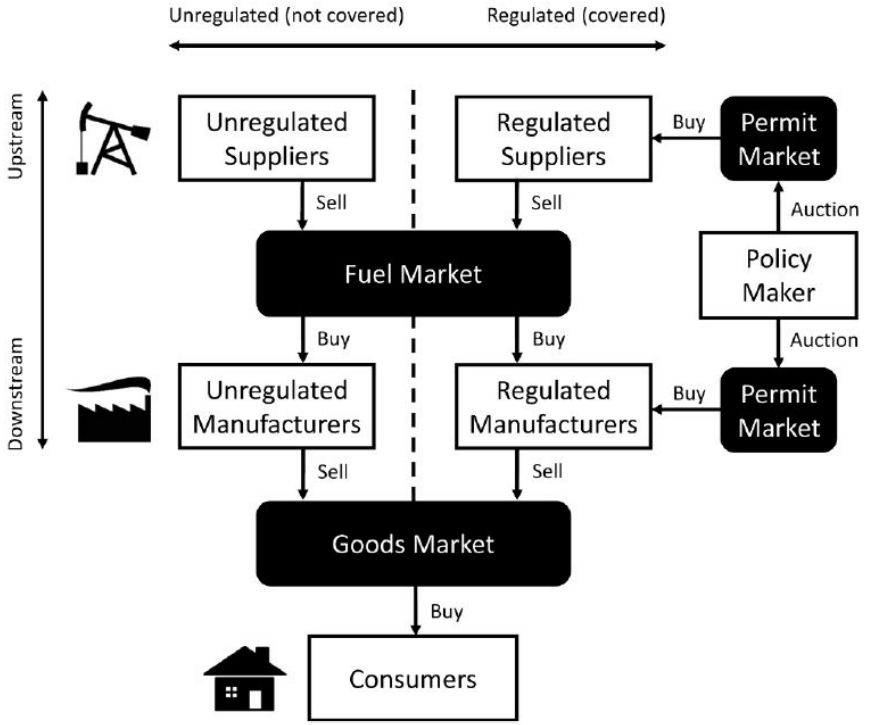
[a] Institute of Environmental Science and Technology, Universitat Autónoma de Barcelona, Spain
[b] Institute for Environmental Studies, Vrije Universiteit Amsterdam, the Netherlands
[c] Graduate School of Economics and Management, Ural Federal University, Yekaterinburg, Russian Federation
[d] ICREA, Barcelona, Spain
[e] School of Business and Economics, Vrije Universiteit Amsterdam, the Netherlands

ARTICLE INFO

Keywords:
Agent-based modeling
Carbon leakage
Carbon tax
Climate policy
Emission trading
Quota

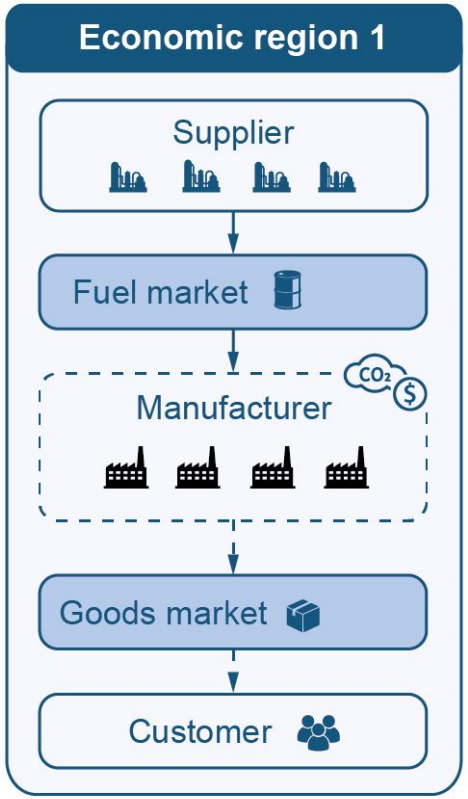
ABSTRACT

Climate policies can be applied either upstream, where fossil fuels are extracted, or downstream, where emissions are generated. Specific policy instruments can be defined for either level, and can take the form of a price signal such as through a tax, or a quantity limit such as through direct regulation or a permit market. In this study, we present an agent-based model to compare the performance of these different instruments and regulation levels. Since policy coverage is often limited, i.e. not all firms being under the regulator's control, we also examine the impact of incomplete coverage on relative policy performance. Our analysis shows that only upstream regulation leads to an increase in fossil fuel prices, which is beneficial under limited coverage as it also affects firms not directly affected by the policy instruments; that prices under quantity-based regulation can decline after an initial peak, stabilizing at a lower level than under the tax; and that direct regulation is more efficient when applied upstream.

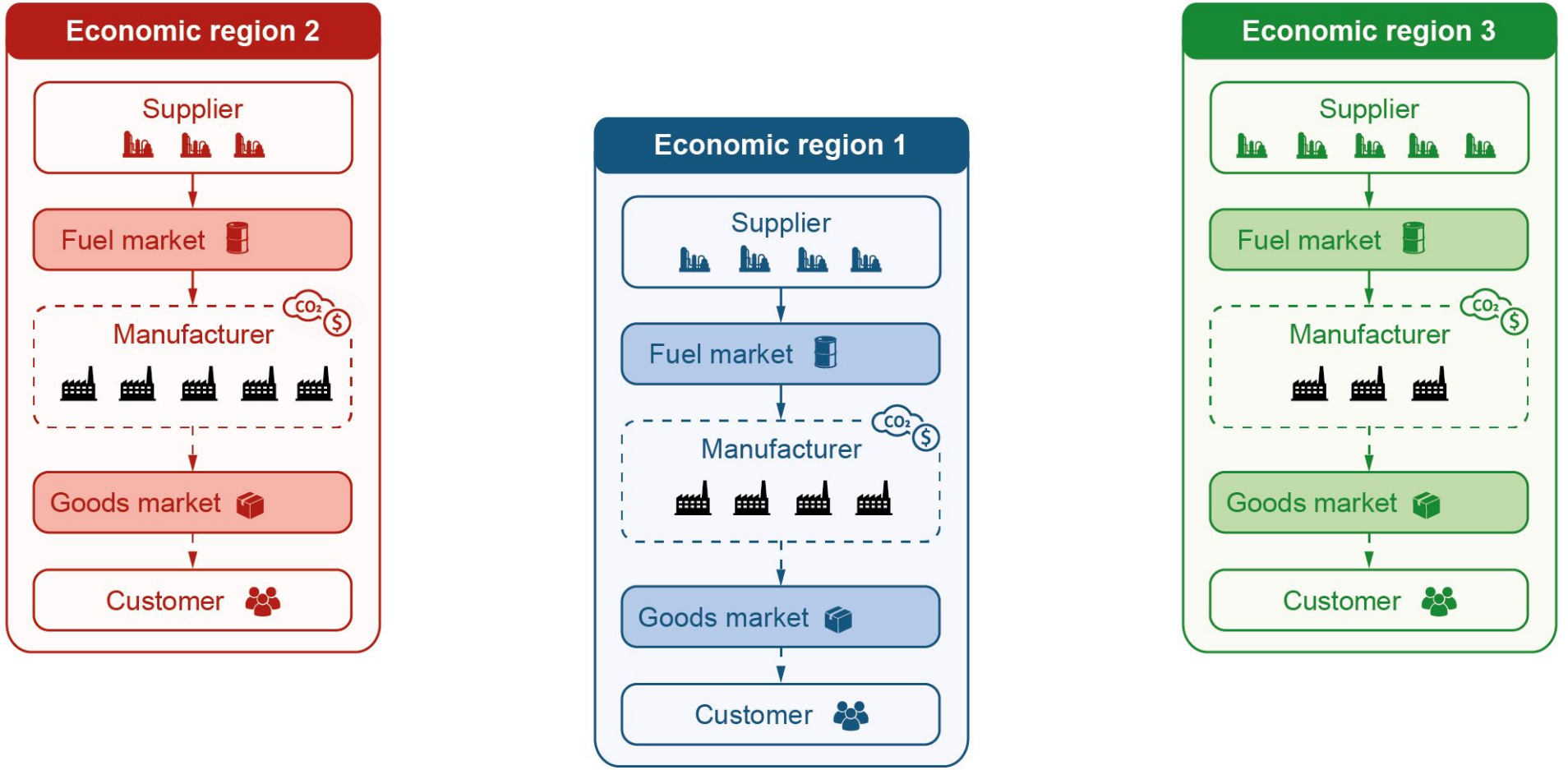


"Regulation at the source? Comparing upstream and downstream climate policies", (https://doi.org/10.1016/j.techfore.2021.121060), Foramitti *et al.*

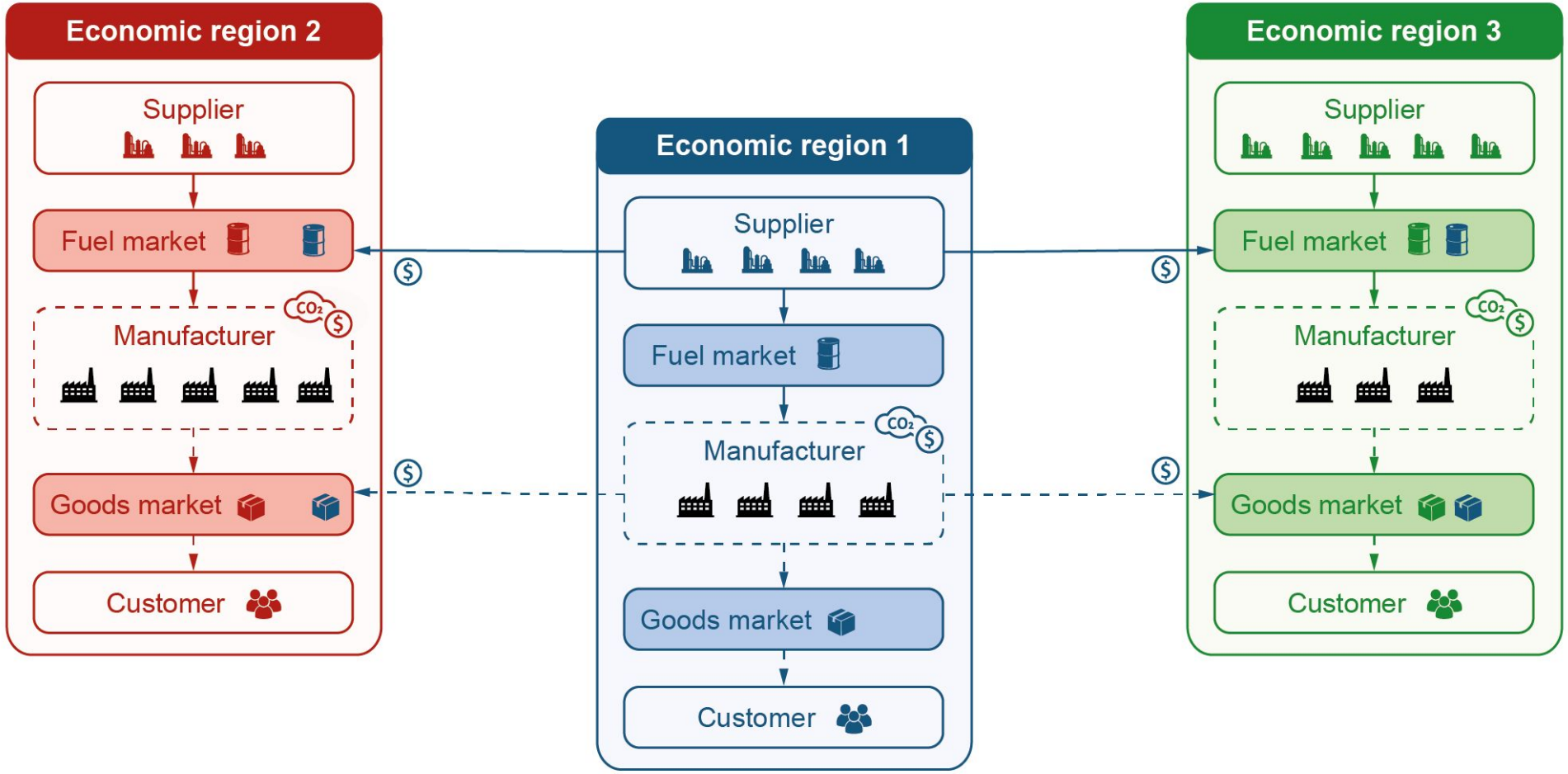
Model description



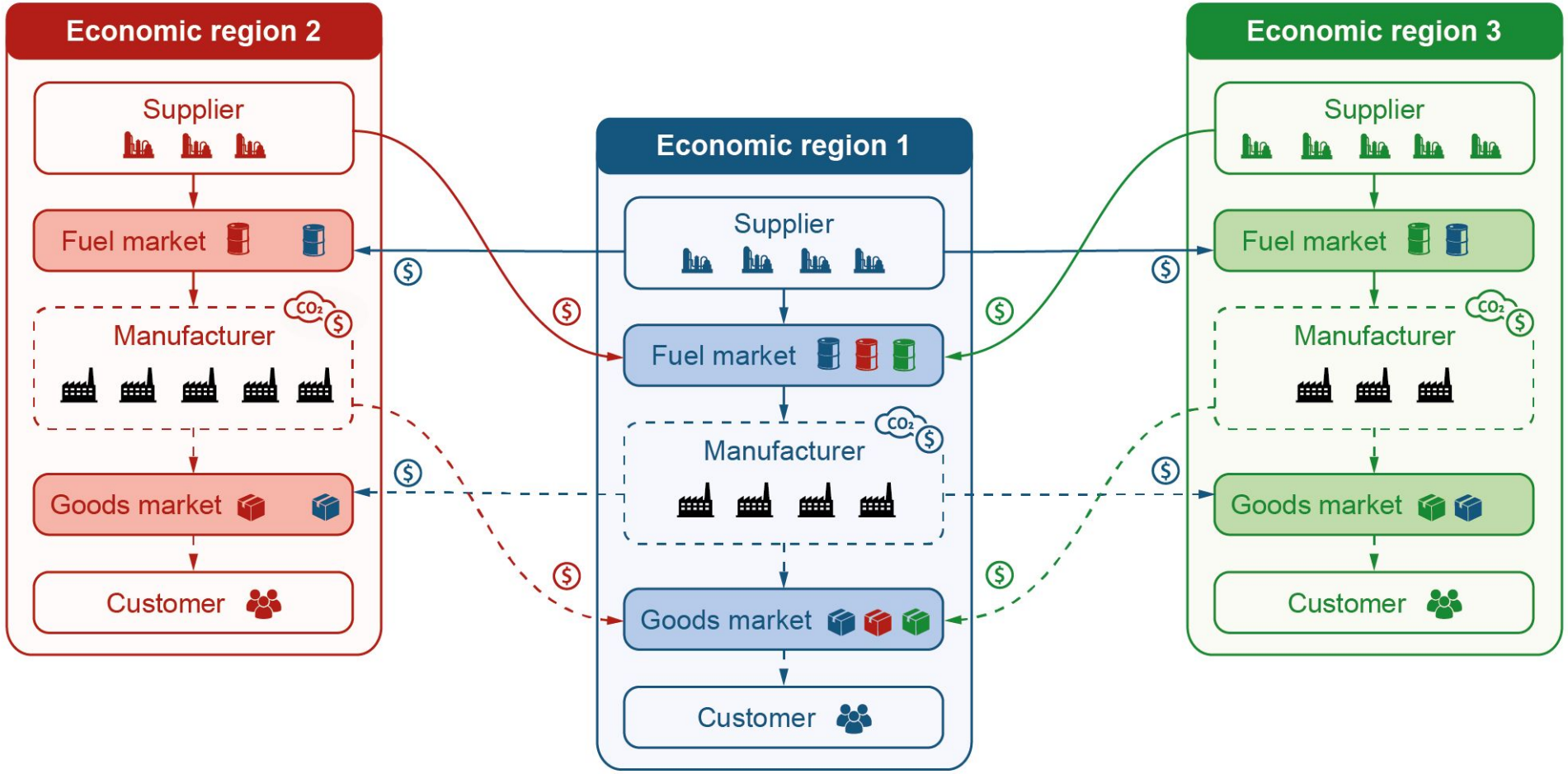
Model description



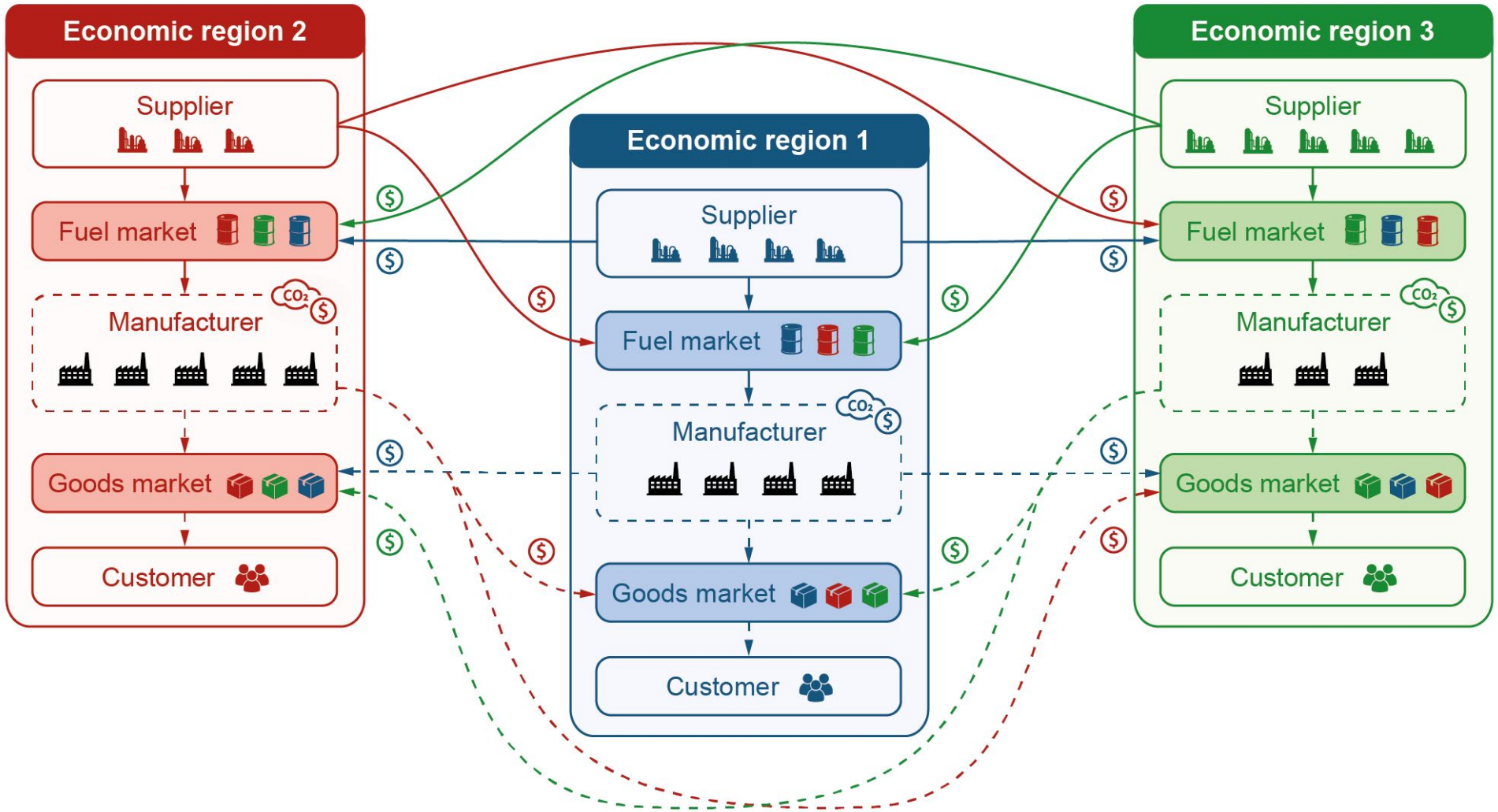
Model description



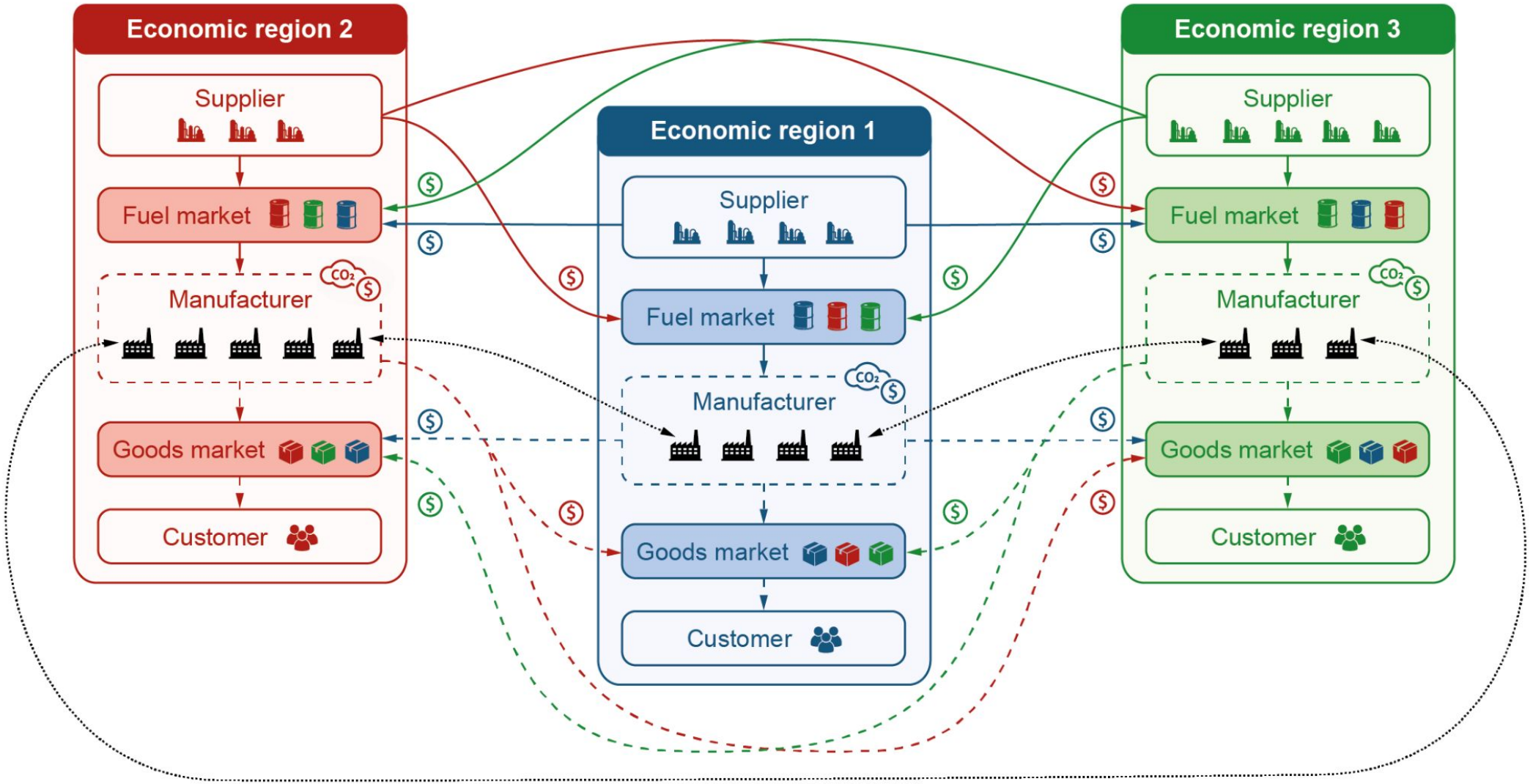
Model description



Model description



Model description



Implementation of carbon tax policies

- We consider *n* = 3 regions with different degree of economic development

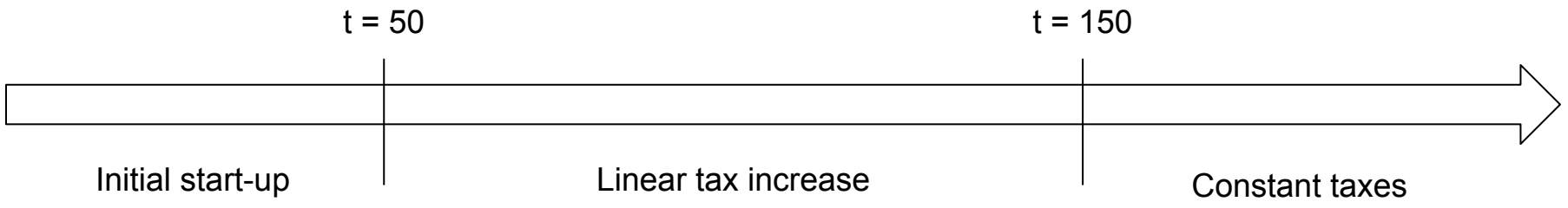
  i ∈ *{developed regions, regions in transition, developing regions}*

- Each region can impose a different level of carbon tax σ_i

  {no tax, low tax, medium tax, high tax}

- Taxes are proportional to the volume of goods produced and the fuel intensity *A*

- All region's carbon taxes go into effect simultaneously:



Manufacturer relocations

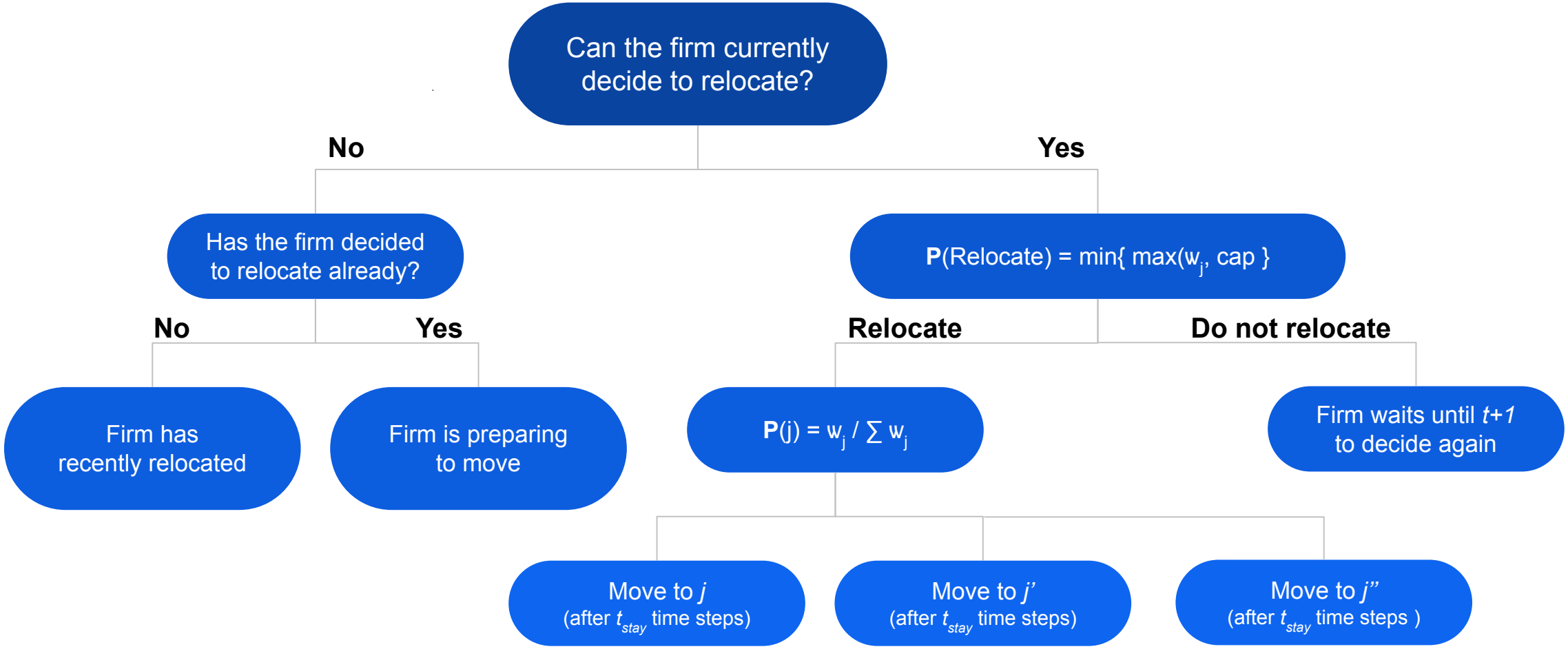
$$\frac{d}{dt}N_j = \sum_{i \neq j} \left(\omega(j \mid i) \cdot N_i - \omega(i \mid j) \cdot N_j \right)$$

$\omega(j \mid i)$: firm's rate of transition moving from region *i* to region *j*

E = profitability = (demand) · (per unit profit margin)

$$\omega(j \mid i) = \gamma \frac{N}{\overline{N}} \cdot \max\left(\frac{E - \overline{E}}{E}, 0\right)$$

Manufacturer relocations at time step *t*

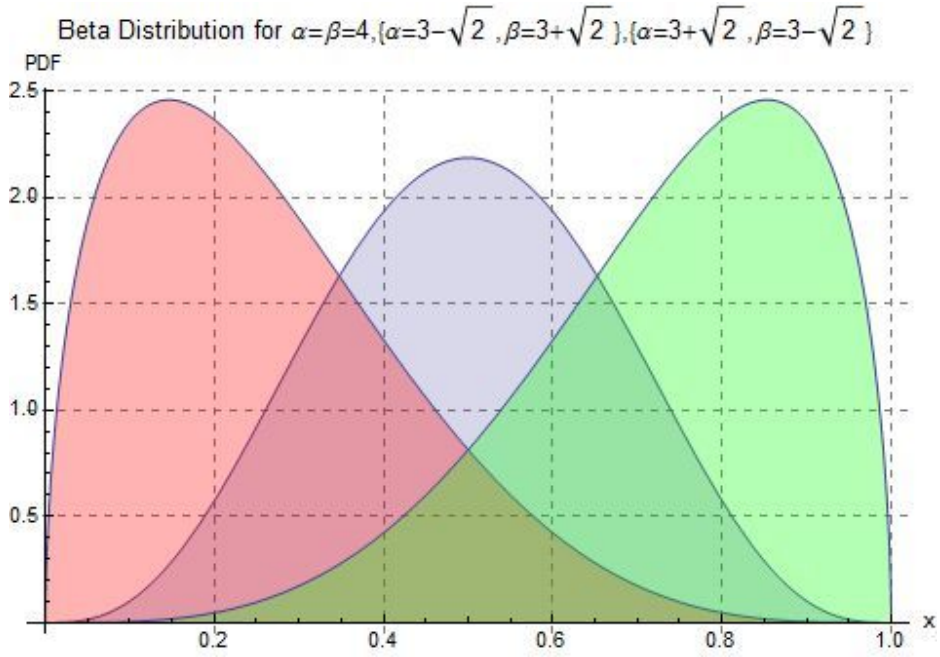


Emissions abatement

$$\text{Marginal cost of abatement (MAC)} = \frac{\text{Additional production cost}}{\text{Reduction in emissions}}$$

MAC + Profitability $\overset{?}{<}$ Carbon tax + Fuel cost

→ Carbon abatement technology adopted



Model implementation details

AgentPy Framework

- Facilitating agent definition

AgentPy - Agent-based modeling in Python

pypi v0.1.5 | license BSD-3-Clause | docs passing | JOSS 10.21105/joss.03065

AgentPy is an open-source library for the development and analysis of agent-based models in Python. The framework integrates the tasks of model design, interactive simulations, numerical experiments, and data analysis within a single environment. The package is optimized for interactive computing with IPython, IPySimulate, and Jupyter.

Please cite this software as follows:

```
Foramitti, J., (2021). AgentPy: A package for agent-based modeling in Python.
Journal of Open Source Software, 6(62), 3065, https://doi.org/10.21105/joss.03065
```

https://github.com/JoelForamitti/agentpy

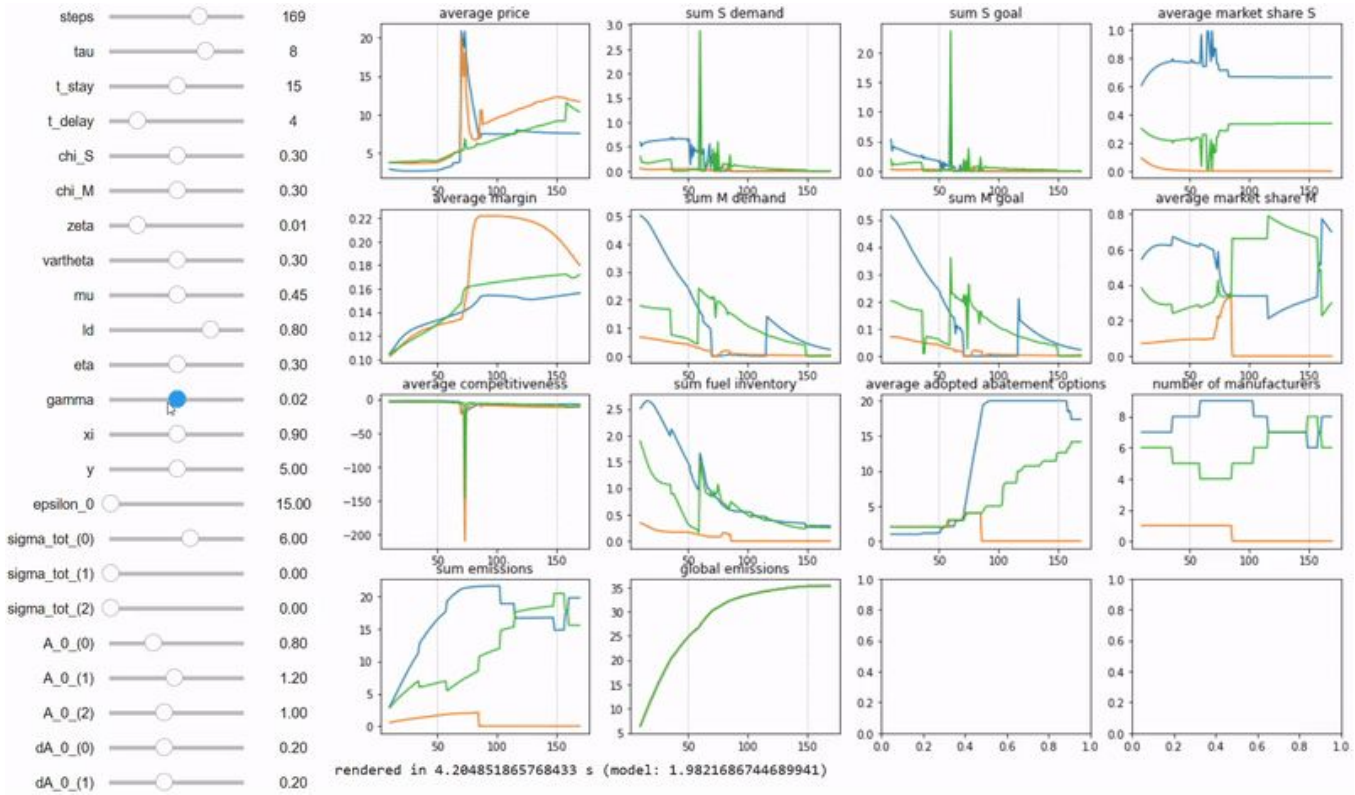
Model implementation details

AgentPy Framework

- Facilitating agent definition

Large parameter space

- Interactive dashboard



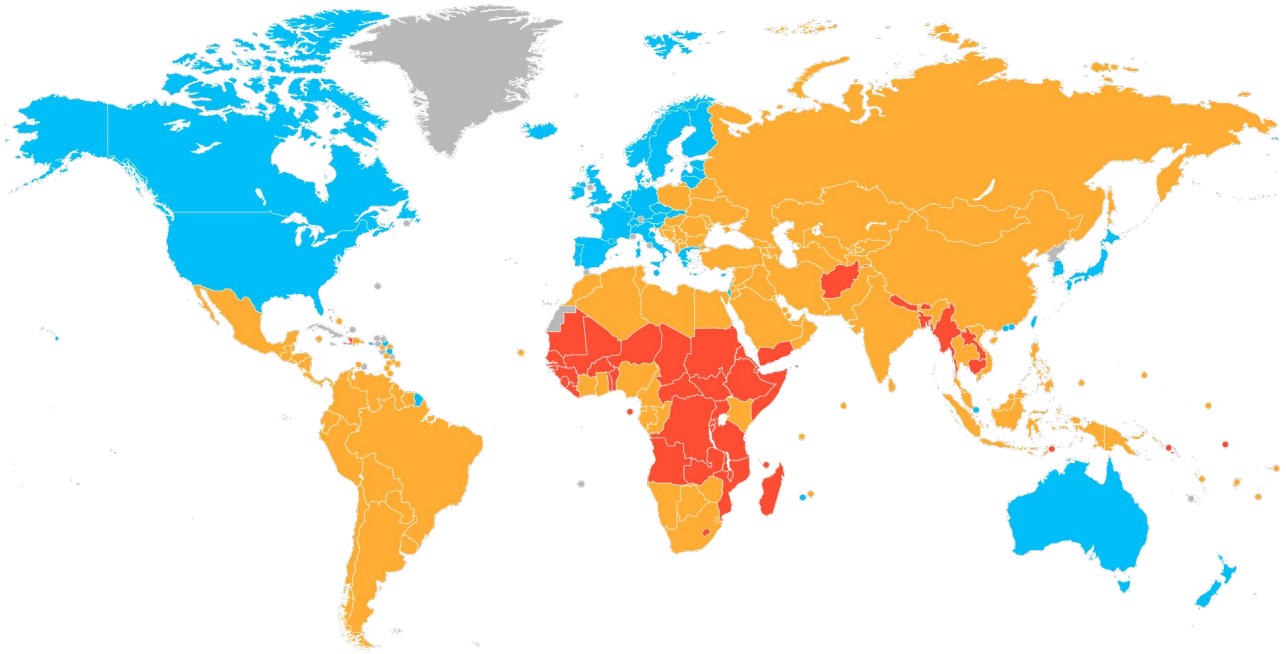
Model implementation details

AgentPy Framework

- Facilitating agent definition

Large parameter space

- Interactive dashboard
- Important parameters backed by data



data source: http://www.unohrlls.org/en/ldc/related/62/

- Oil production
- GDP
- Global manufacturing output

Model implementation details

AgentPy Framework

- Facilitating agent definition

Large parameter space

- Interactive dashboard
- Important parameters backed by data

Multi-run experiment

- 100 runs of different scenarios

Scenario	Region		
	Developed	Transitioning	Developing
All No Tax	None	None	None
All Low Tax	Low	Low	Low
All High Tax	High	High	High
High-Low-Low	High	Low	Low

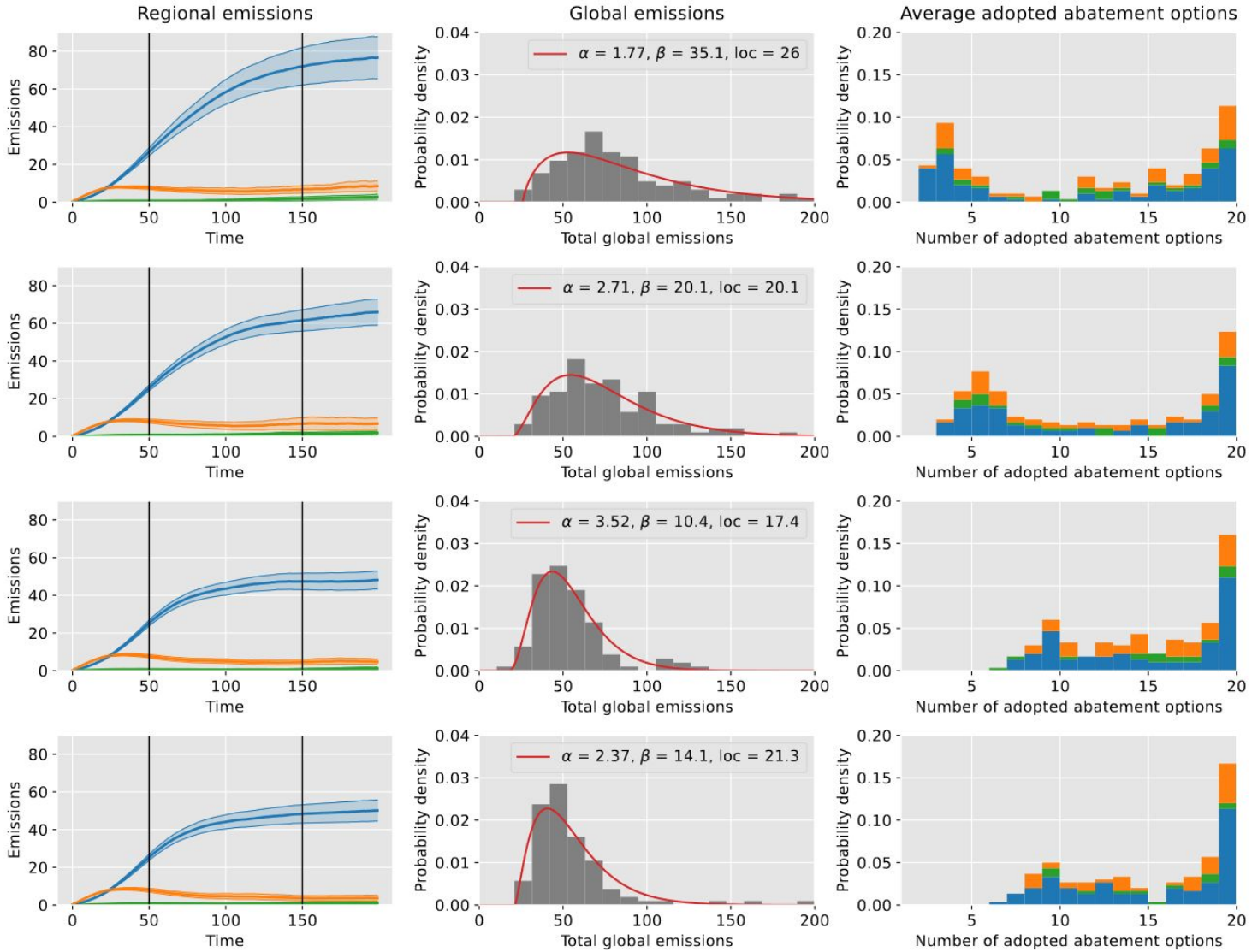
Multi-run results

All No Tax

All Low Tax

All High Tax

High-Low-Low Tax



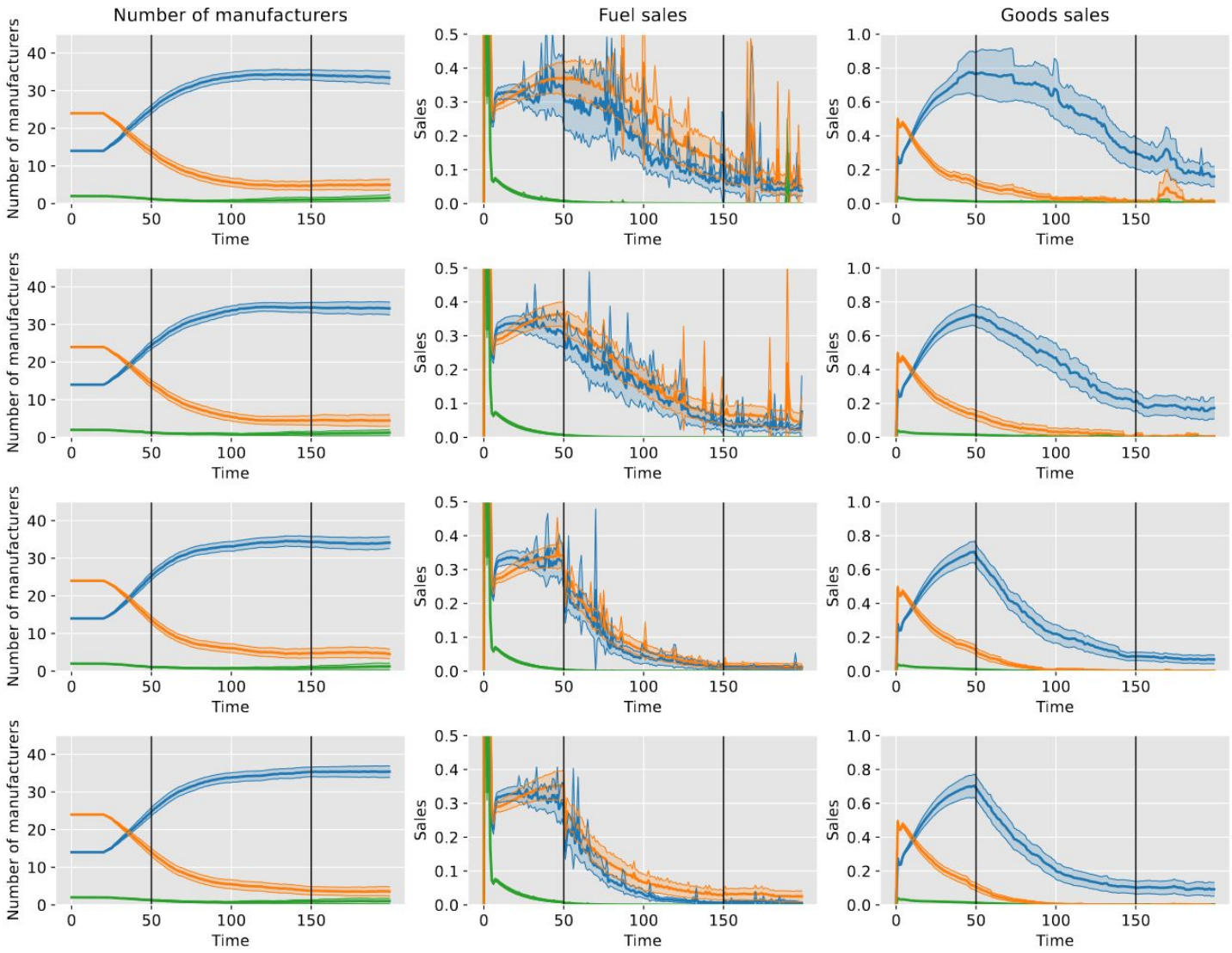
Multi-run results

All No Tax

All Low Tax

All High Tax

High-Low-Low Tax



Sensitivity analysis

SALib

- Sensitivity index: First-order indices (the contribution to the output variance by a *single* model input)

- Sampling method: Saltelli's extension of the Sobol' sequence (*uniform* samples of the parameter space)

SALib - Sensitivity Analysis Library in Python

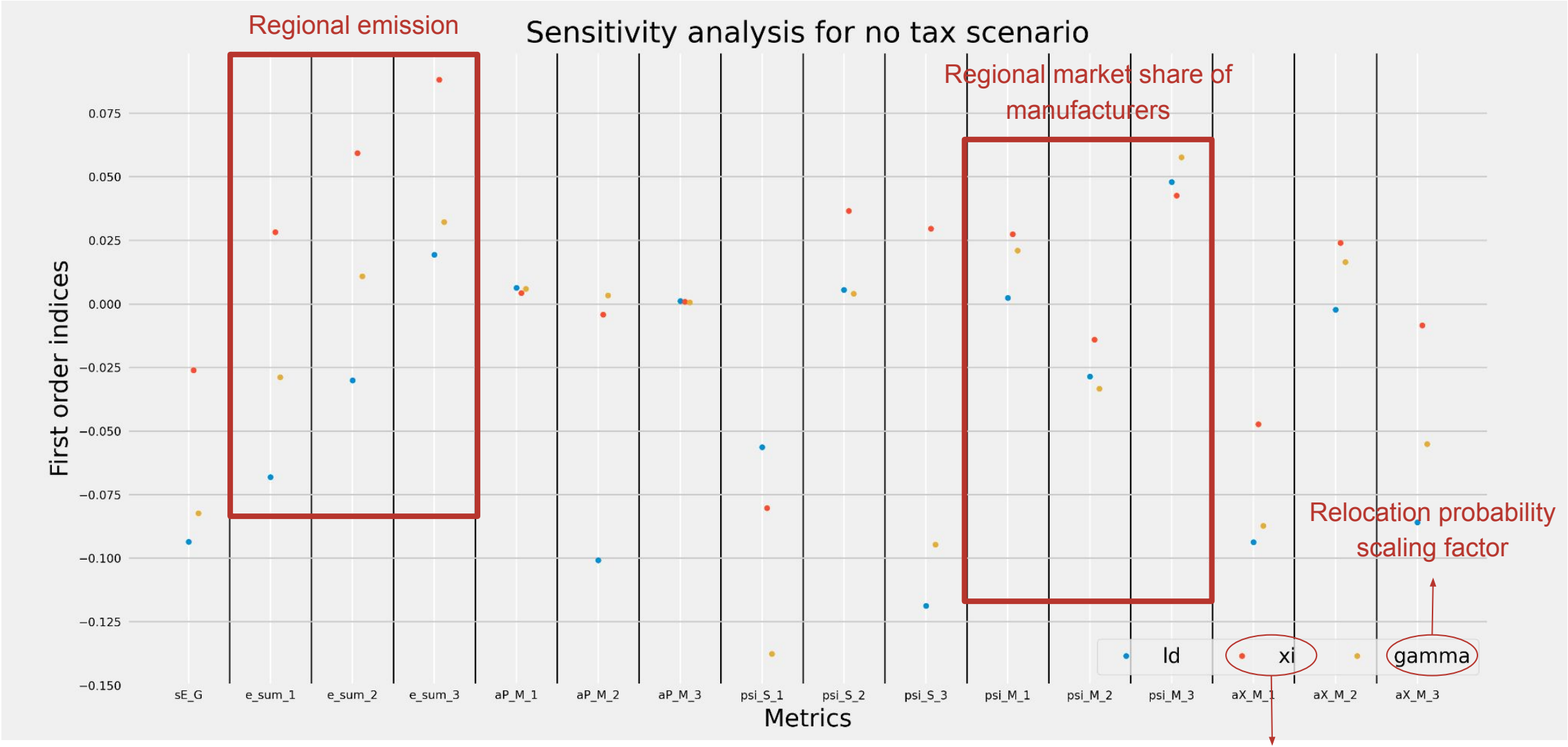
DOI 10.5281/zenodo.160164 JOSS 10.21105/joss.00097

Python implementations of commonly used sensitivity analysis methods, including Sobol, Morris, and FAST methods. Useful in systems modeling to calculate the effects of model inputs or exogenous factors on outputs of interest.

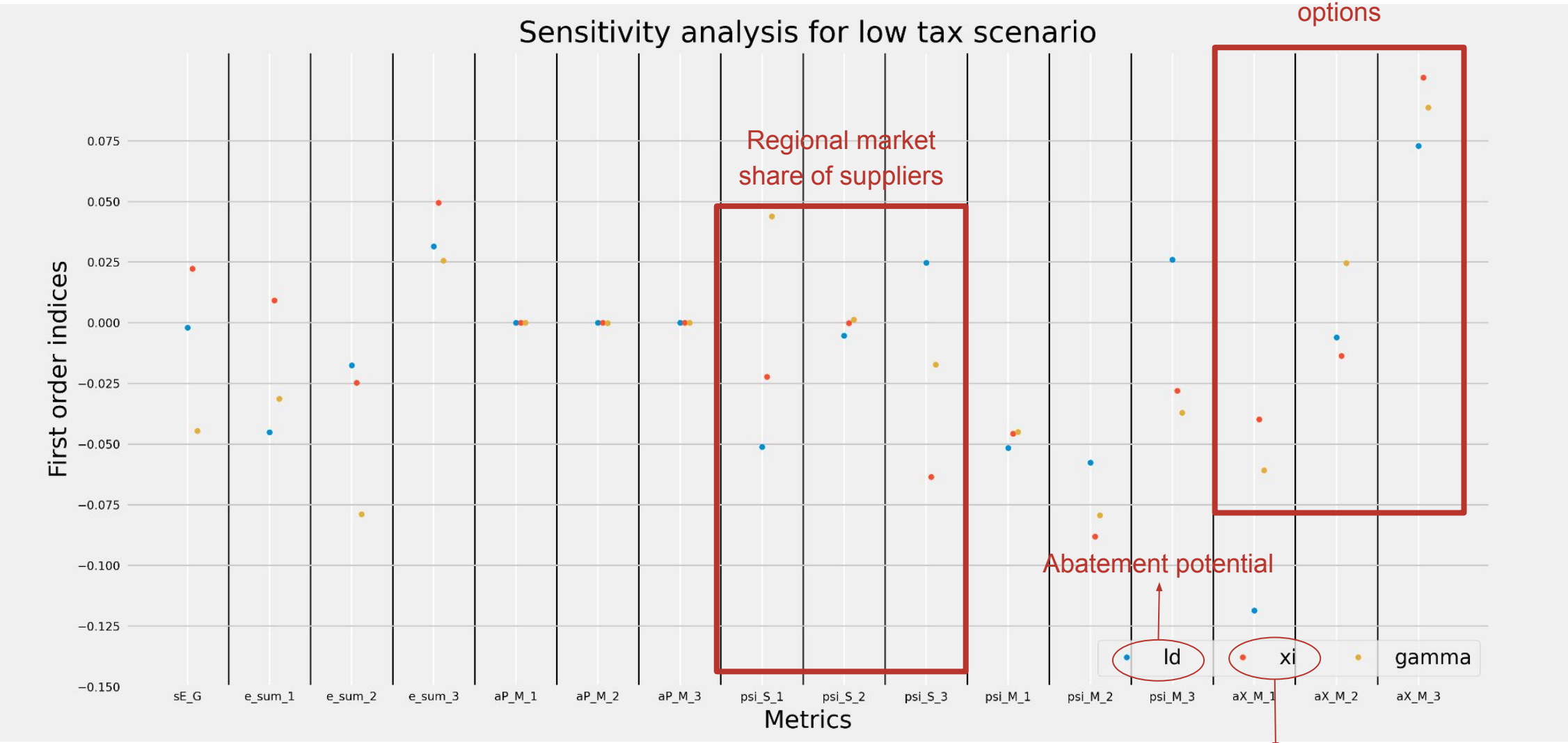
Supported Methods

- Sobol Sensitivity Analysis (Sobol 2001, Saltelli 2002, Saltelli et al. 2010)
- Method of Morris, including groups and optimal trajectories (Morris 1991, Campolongo et al. 2007)
- Fourier Amplitude Sensitivity Test (FAST) (Cukier et al. 1973, Saltelli et al. 1999)
- Random Balance Designs - Fourier Amplitude Sensitivity Test (RBD-FAST) (Tarantola et al. 2006, Elmar Plischke 2010, Tissot et al. 2012)
- Delta Moment-Independent Measure (Borgonovo 2007, Plischke et al. 2013)
- Derivative-based Global Sensitivity Measure (DGSM) (Sobol and Kucherenko 2009)
- Fractional Factorial Sensitivity Analysis (Saltelli et al. 2008)
- High Dimensional Model Representation (Li et al. 2010)

Sensitivity analysis



Sensitivity analysis



Outlook

- Calibrate the model with more accurate data to reflect the current global economy
- Include more diverse climate policies like carbon permit trading, carbon quotas
- Consider the entry and exit of firms
- Add industry-specific features to increase the heterogeneity of firms

ETH zürich

Thank you!

Modelling the impact of carbon tax policies across multiple economic regions

Wen Yi Chan, Joel Huber, Gaspard Krief, Tianyi Liu, Pascal Troxler