Optimization of Trading Strategies in Continuous Intraday Markets for a Storage Unit

Gilles Bertrand, Anthony Papavasiliou

Center for Operations Research and Econometrics
Université catholique de Louvain

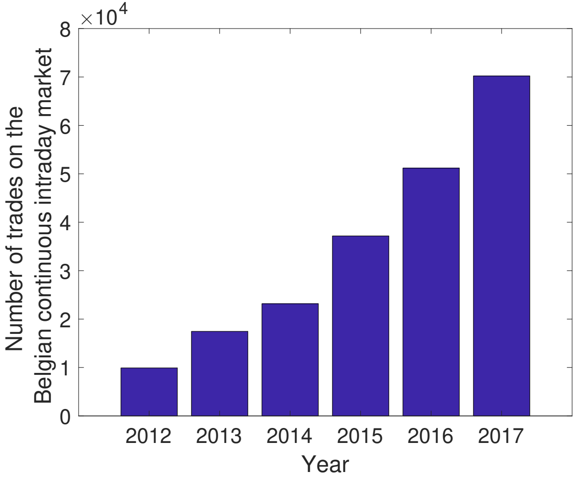
December 7, 2020

Outline

Outline

1. Continuous Intraday Market Description
2. Threshold Policy
3. Generalization of the Threshold Policy
4. Case Study: German Continuous Intraday Market

Motivation


×10^4
8
7
6
5
4
3
2
1
0
Number of trades on the
Belgian continuous intraday market
2012
2013
2014
2015
2016
2017
Year

Description of the Continuous Intraday Market

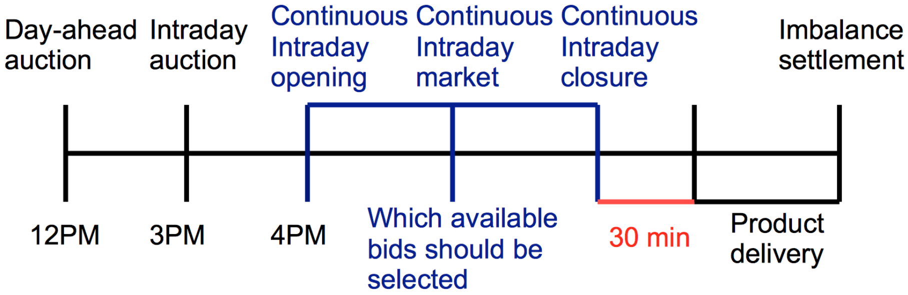


Figure: Short-term German electricity market

Format of Intraday Bids

	Hour	Type	Price (€/MWh)	Quantity (MW)
Bid 1	1	s	28	10
Bid 2	1	b	25	5
Bid 3	1	b	30	8
Bid 4	1	b	25	2.5
Bid 5	1	s	27	0.3
Bid 6	2	b	29	0.8
Bid 7	14	s	32	3

- Bids arrive continuously in the intraday platform
- Bids are reserved on first-come-first-serve basis

Literature Review

Intraday price models

- [Kiesel 2015]: Econometric study of the parameters influencing the price evolution
- [Kiesel 2017]: modelling of order arrivals using Hawkes process

Trading by assuming a price model

- [Aid 2015]: solving the trading problem of a thermal generator using stochastic differential equations, assuming some model for the price evolution
- [Braun 2016]: solving the problem of optimizing pumped storage trading if we have access to a price curve for the coming hours

Trading without assuming a price model

- [Skajaa 2015]: heuristic method for covering the position of a wind farm based on imbalance price forecast

Our Goal

We are interested in a *model-free* approach that can handle

- continuous arrival of orders
- multi-stage uncertainty
- management of flexible (e.g. pumped hydro, storage) assets

Assumptions

1. The trading strategies that we develop are balanced
2. We only accept bids that are already present in the market
3. We do not consider block or integer bids
4. We assume that our strategy does not influence the future bids of the other participants

Outline

Definition of a Markov Decision Process

Markov decision process

A Markov decision process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where

- $\mathcal{S}$ is a set of states
- $\mathcal{A}$ is a set of actions
- $\mathcal{R}$ is a reward function, $\mathcal{R}(s, a) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$
- $\mathcal{P}^a_{s,s'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$ is the probability to arrive in state s' if we follow action a in state s

Definition of a Markov Decision Process

Objective function

We optimize over a set of policies for the sum of reward if we follow a policy

$$\max_{\pi \in \Pi} \sum_{t=1}^{T} \mathbb{E}\left[R_t(S_t, A^{\pi}(S_t))\right]$$

Policy Function Approximation

Policy function approximation (PFA)

The idea in PFA is to approximate directly the policy

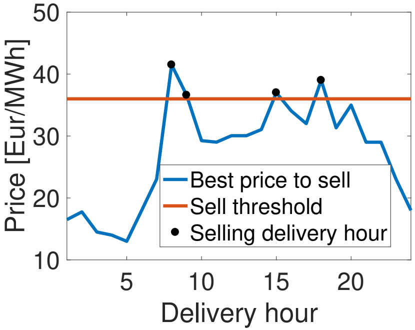
$$\pi(a|s;\theta) = \mathbb{P}[A_t = a|S_t = s;\theta]$$

Policy Function Approximation

Policy function approximation (PFA)

The idea in PFA is to approximate directly the policy

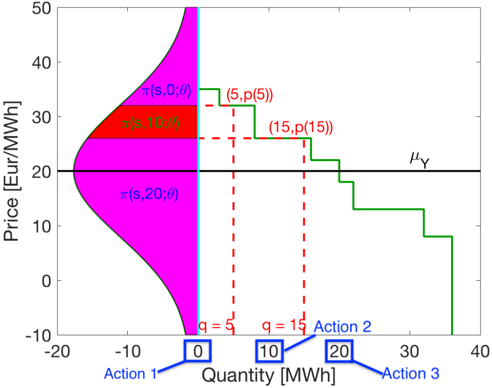
$$\pi(a|s; {\color{red}\theta}) = \mathbb{P}[A_t = a | S_t = s; {\color{red}\theta}]$$



Threshold policy

Threshold policies have been *proven* to be optimal for specific instances of stochastic optimal control problems with uncertain prices in [Morris], [Golabi], [Kingsman].

Graphical Representation of Threshold Policy for Pumped Hydro Problem



Graphical Representation of Threshold Policy

We use a *threshold policy*, which is a distribution over actions:

- The bell curve indicates the probability density function of the sell threshold
- The two purple segments and the red segment of the bell curve indicate the probability of each of the three actions:
  - Sell 0 MWh
  - Sell 10 MWh
  - Sell 20 MWh
- The green decreasing function corresponds to the buy bids that are available in the order book for a given trading hour

We are interested in finding an **optimal** threshold

REINFORCE Algorithm

Algorithm

REINFORCE algorithm for finite horizon:

- Initialize θ_0
- for each episode $\{s_1, a_1, r_2, \cdots, s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$

  for t $= 1 : T - 1$ do

  $$\theta = \theta + \alpha \nabla_\theta \log(\pi(s_t, a_t; \theta)) g_t$$

  end for

  end for

Remark

g_t is the profit from t to the end T of the episode

$\nabla_\theta \log$

These gradients can be expressed in closed form

Outline

Generalization of the Threshold Policy

Let $f\colon R^n \to R$ be a differentiable function s.t. $\theta = f(\alpha)$. We can compute the derivative with respect to α by using the chain rule:

$$\begin{aligned} \frac{\partial \pi(s;\theta)}{\partial \alpha} &= \frac{\partial \pi(s;\theta)}{\partial \theta}\frac{\partial \theta}{\partial \alpha} \\ &= \frac{\partial \pi(s;\theta)}{\partial \theta}\frac{\partial f}{\partial \alpha} \end{aligned}$$

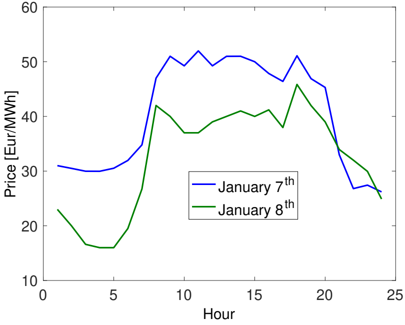
This allows us to influence the threshold by observing relevant factors

Expected Behaviour of a Threshold Policy

1. Adapt with respect to the intraday auction price
2. Adapt with respect to the delivery time
3. Adapt with respect to the evolution of intraday prices
4. Adapt with respect to the remaining time
5. Adapt with respect to the price of the other delivery times
6. Adapt with respect to round-trip efficiency
7. Ensure that the stored volume respects reservoir limits

Goal 1: Adapting to Intraday Auction Price

- The same threshold for every day would give bad results because the price level can be different from day to day (left graph)
- The intraday auction price seems as a good explanatory variable (right graph)



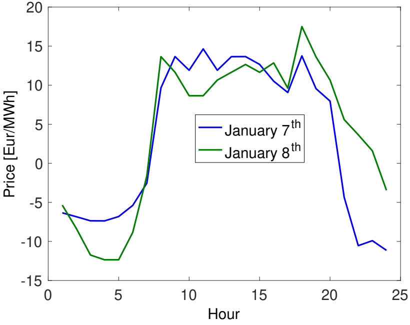


Figure: Price for two different days (left). Price for two different days at which we subtract the average intraday auction price (right).

Goal 1: Adapting to Intraday Auction Price

- The buy threshold is taken as the minimum value of the intraday curve at which we add a security margin that will be learned by the algorithm.
- The sell threshold is taken as the maximum value of the intraday curve at which we subtract a security margin that will be learned by the algorithm.

$$\mu_X \leftarrow p_{\min} + \alpha_1^s (p_{\max} - p_{\min})$$
$$\mu_Y \leftarrow p_{\max} - \alpha_1^b (p_{\max} - p_{\min})$$

Goal 1: Adapting to Intraday Auction Price

What we decide

We decide the parametrization of the policy. We start at p_{min} and add a security margin $\alpha_1^s(p_{\text{max}} - p_{\text{min}})$

What the algorithm decide

This security margin α_1^s is learned through experience by the REINFORCE algorithm.

Outline

Case Study

- Data source: 2 years of data of the German CIM, procured from EPEX
- Training data: 200 days of 2015
- Testing data: 531 remaining days of 2015 and 2016

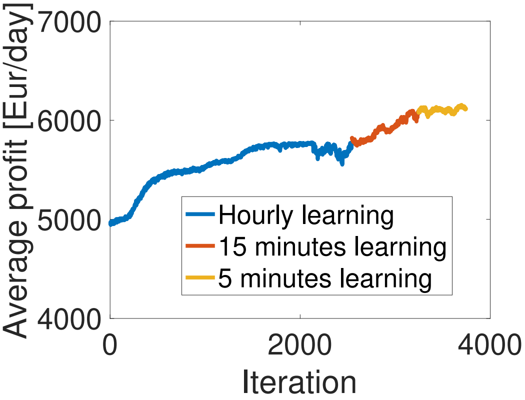
Influence of the Frequency

Length of time step	Percentage of offers observed
1 hour	**25.7**
15 minutes	41.5
5 minutes	56
1 minute	**74.8**
15 seconds	86.7
5 seconds	92.2
1 second	**98.3**

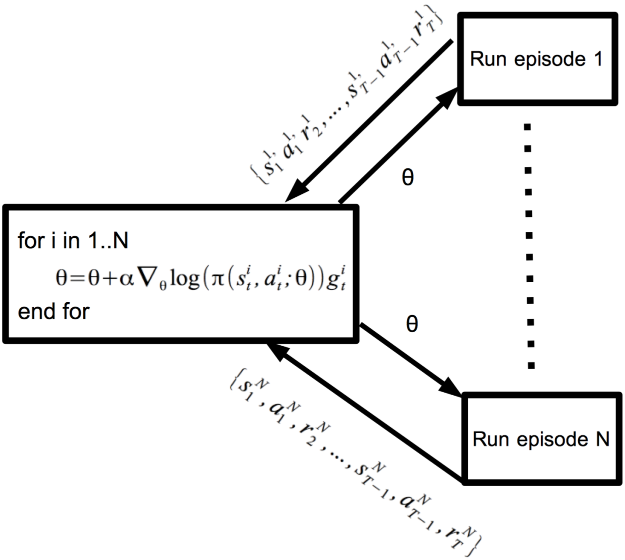
- By trading at hourly frequency, 75% of the data would never be observed
- If we trade every minute, we have more than 1000 stages which is intractable for stochastic programming methods such as SDDP
- In our case, we trade every second when we are testing

Training the Policy Function

This graph shows the evolution of the *profit* with respect to the *iteration*. An *iteration* corresponds to 3 repetitions of our 200 days of learning. These 600 episodes are run using parallel computing.

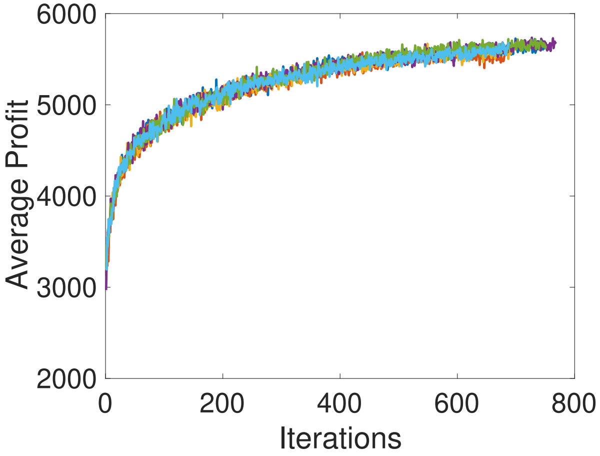


Parallel implementation



Stability of the learning phase

Evolution of the daily profit for 6 realizations of the REINFORCE algorithm



Competing Policies

We consider the *rolling intrinsic* policy as a benchmark [Lohndorf, Wozabal, 2015]

- Applied for intraday trading with pumped hydro
- Receding horizon approach
- Myopic: accept any feasible trade that gives an *instantaneous* profit

Competing Policies

We have compared the results of three different methods:

- **Rolling intrinsic 4pm:** rolling intrinsic method launched at 4pm
- **Rolling intrinsic 11pm:** rolling intrinsic method launched at 11pm
- **Threshold:** our proposed threshold policy

Comparison of Policies

Trading frequency	Method	Efficiency	Used data	Profit mean [€/day]
1 hour	Threshold	1	out	5374
1 hour	Rolling 11PM	1	out	4591
1 second	Threshold	1	out	**6405**
1 second	Rolling 11PM	1	out	**5438**
1 second	Rolling 4PM	1	out	4742
1 second	Threshold	0.81	out	3762
1 second	Rolling 11PM	0.81	out	3311
1 second	Threshold	1	in	6605
1 second	Rolling 11PM	1	in	5694

Our threshold policy performs better than rolling intrinsic

Comparison of Policies

Trading frequency	Method	Efficiency	Used data	Profit mean [€/day]
1 hour	Threshold	1	out	5374
1 hour	Rolling 11PM	1	out	4591
1 second	Threshold	1	out	6405
1 second	Rolling 11PM	1	out	5438
1 second	Rolling 4PM	1	out	4742
1 second	Threshold	0.81	out	3762
1 second	Rolling 11PM	0.81	out	3311
1 second	Threshold	1	in	6605
1 second	Rolling 11PM	1	in	5694

The profit of both methods increases with the trading frequency

Comparison of Policies

Trading frequency	Method	Efficiency	Used data	Profit mean [€/day]
1 hour	Threshold	1	out	5374
1 hour	Rolling 11PM	1	out	4591
1 second	Threshold	1	out	6405
1 second	Rolling 11PM	1	out	5438
1 second	Rolling 4PM	1	out	4742
1 second	Threshold	0.81	out	3762
1 second	Rolling 11PM	0.81	out	3311
1 second	Threshold	1	in	6605
1 second	Rolling 11PM	1	in	5694

Rolling intrinsic profit is less good if we start the method at 4PM

Comparison of Policies

Trading frequency	Method	Efficiency	Used data	Profit mean [€/day]
1 hour	Threshold	1	out	5374
1 hour	Rolling 11PM	1	out	4591
1 second	Threshold	1	out	6405
1 second	Rolling 11PM	1	out	5438
1 second	Rolling 4PM	1	out	4742
1 second	Threshold	0.81	out	3762
1 second	Rolling 11PM	0.81	out	3311
1 second	Threshold	1	in	6605
1 second	Rolling 11PM	1	in	5694

Our method is also efficient with an imperfect round-trip efficiency

Comparison of Policies

Trading frequency	Method	Efficiency	Used data	Profit mean [€/day]
1 hour	Threshold	1	out	5374
1 hour	Rolling 11PM	1	out	4591
1 second	Threshold	1	out	6405
1 second	Rolling 11PM	1	out	5438
1 second	Rolling 4PM	1	out	4742
1 second	Threshold	0.81	out	3762
1 second	Rolling 11PM	0.81	out	3311
1 second	Threshold	1	in	6605
1 second	Rolling 11PM	1	in	5694

Our method achieves a robust performance against out-of-sample data

Comparison of Policies

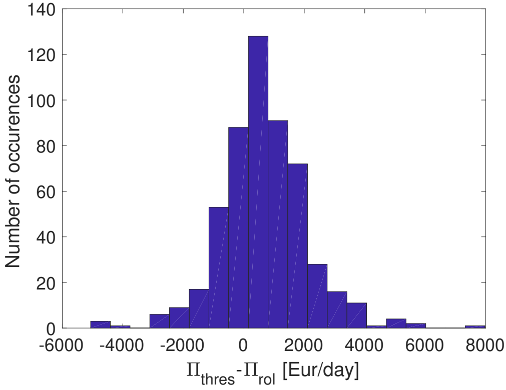
Trading frequency	Method	Efficiency	Used data	Profit mean [€/day]
1 hour	Threshold	1	out	5374
1 hour	Rolling 11PM	1	out	4591
1 second	Threshold	1	out	**6405**
1 second	Rolling 11PM	1	out	**5438**
1 second	Rolling 4PM	1	out	4742
1 second	Threshold	0.81	out	3762
1 second	Rolling 11PM	0.81	out	3311
1 second	Threshold	1	in	6605
1 second	Rolling 11PM	1	in	5694

In the next slides, we will compare the two best performing policies:

- rolling intrinsic starting at 11pm
- threshold policy

Distribution of Profits

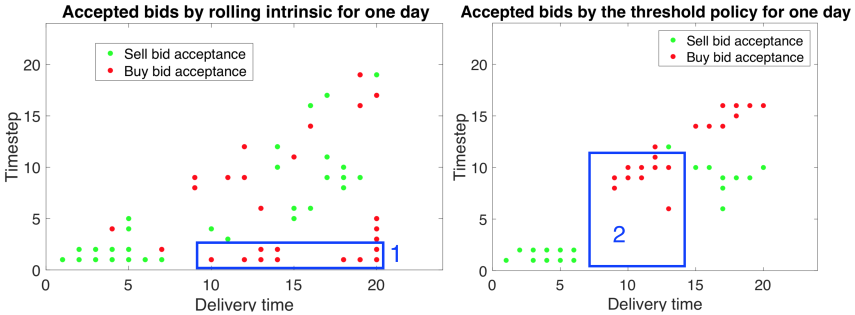
- One *occurrence* corresponds to one day of trading
- The profit is accumulated gradually and is not coming from one spike



Different Attitude towards Risk

There is a trade-off between

1. arbitraging against earlier bids with less interesting prices (rolling intrinsic, risk-free)
2. waiting for more interesting prices later in the day (threshold policy, more risky)



Conclusions and Perspectives

- Observations
  - Our method outperforms rolling intrinsic with statistical significance
  - The profit of rolling intrinsic varies significantly with the time that trading commences
- Future research
  - Adapt the methodology for a wind farm
  - Compare our RL approach with SDDP

Thank you

Contact:

Gilles Bertrand, gilles.bertrand@uclouvain.be
https://sites.google.com/site/gillesbertrandresearch/home

Anthony Papavasiliou, anthony.papavasiliou@uclouvain.be
http://perso.uclouvain.be/anthony.papavasiliou/

Appendix

- Notation
- MDP for hydro
- REINFORCE algorithm analyses
- Computing ∇_θlog
- Respecting reservoir limits
- Adapting to intraday auction price
- Graph of the α

Notation

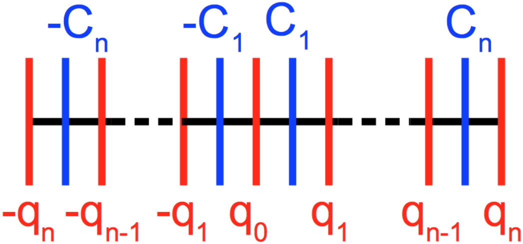
- Sets
  - T: set of time steps
  - D: set of delivery times
  - I_d: set of bids *available* for delivery time d
- Variables
  - $q_{it}^{s/b}, i \in I_d, d \in D, t \in T$: quantity of sell/buy bid i for delivery time d at timestep t which the agent accepts
  - $v_{t,d}, t \in T, d \in D$: quantity of energy that would be stored at delivery time d in the reservoir if we do not trade after time step t

Notation

- Parameters
  - v_{0d}: initial reservoir volume we would get in d *given* trade in previous markets
  - V: maximum energy that can be stored in the reservoir
  - $P_i^{s/b}, i \in I_d, d \in D$: the price of sell/buy bid i for delivery time d
  - $Q_{it}^{s/b}, i \in I_d, d \in D$: the quantity of sell/buy bid i for delivery time d that is available at time step t

Aggregation of Orders

In order to keep the size of the state space tractable, we *aggregate* orders



- q_i: are the discrete quantities (in MW) that we can trade
- C_i: mid-points between options

MDP Representation of Continuous Intraday Trading

State variables

- S_t^1: price and revenue for the aggregated quantities
- S_t^2: contracts signed in the past
  - $\mathrm{v}_{t,d}$: quantity that would be stored at delivery time d with the trades accepted up to time step t
- S_t^3: *exogenous* data
  - Intraday auction price
  - Time before delivery

MDP Representation of Continuous Intraday Trading

Decision variables

- We *aggregate* the trading action space into one variable $a_{t,d}$ that indicates how much we sell:

$$a_{t,d} \in \{-q_n, \cdots, -q_1, 0, q_1, \cdots q_n\}$$

Dynamic transition

- $S^2_{t,d}$ evolves according to the following transition equation:

$$v_{t,d} = v_{t-1,d} - \sum_{b \in D | b \leq d} a_{t,b}, \ \forall d \in D$$

MDP Representation of Continuous Intraday Trading

Reservoir limits

$$v_{t,d} \geq 0,\ \forall d \in D$$
$$v_{t,d} \leq V,\ \forall d \in D$$

Objective

$$R_t = \sum_{d \in D} \mathrm{rev}_{t,d}(a_{t,d})$$

where $\mathrm{rev}_{t,d}(a_{t,d})$ is the revenue obtained from the continuous intraday market if we sell $a_{t,d}$ at delivery time d at time step t

REINFORCE Algorithm

Algorithm

REINFORCE algorithm for finite horizon:

- Initialize θ_0
- for each episode $\{s_1, a_1, r_2, \cdots, s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$

  for t = 1 : $T - 1$ do

  $$\theta_{k+1} = \theta_k + \alpha \nabla_\theta \log(\pi(s_t, a_t; \theta)) g_t$$

  end for

end for

Remark

g_t is the profit from t to the end T of the episode

Theoretically

[Sutton, 2018]: this update is equivalent to using a stochastic gradient method on the expected reward $v_{\pi_\theta}(s_t)$

Reinforce algorithm intuition

Intuition

$$\nabla_\theta \log(\pi(s, a; \theta)) r = \frac{\nabla_\theta \pi(s, a; \theta)}{\pi(s, a; \theta)} r$$

- $\nabla_\theta \pi(s, a; \theta)$: direction in which the probability of taking action a in state s is the most increased.
- r: the higher profit we get, the more we want to increase the probability of taking action a in state s.
- $\frac{1}{\pi(s,a;\theta)}$: we want to increase more the probability if the probability was low. It compensates the fact that some states are less observed.

Computing $\nabla_\theta \log$

- In order to implement the REINFORCE algorithm, we require $\nabla_\theta \log(\pi(s_t, a_t; \theta))$
- We can compute the policy in closed form, e.g.

$$\begin{aligned}
\pi(s, -q_n; \theta) &= \Pr(p(-C_n) \leq X) \\
&= 1 - F_X(p(-C_n)) \\
\pi(s, q_n; \theta) &= \Pr(\max(X, Y) \leq p(C_n)) \\
&= F_X(p(C_n))F_Y(p(C_n))
\end{aligned}$$

Computing $\nabla_\theta \log$

- We can also compute the gradients in closed form, e.g.

$$
\begin{aligned}
\frac{\partial \pi(s, -q_i; \theta)}{\partial \mu_X} &= \frac{\partial (F_X(p(-C_{i+1})) - F_X(p(-C_i)))}{\partial \mu_X} \\
&= f_X(p(-C_i)) - f_X(p(-C_{i+1})) \\
\frac{\partial \pi(s, -q_i; \theta)}{\partial \sigma_X} &= \frac{\partial (F_X(p(-C_{i+1})) - F_X(p(-C_i)))}{\partial \sigma_X} \\
&= \frac{\mu_X - p(-C_i)}{\sigma_X} f_X(p(-C_i)) \\
&- \frac{\mu_X - p(-C_{i+1})}{\sigma_X} f_X(p(-C_{i+1}))
\end{aligned}
$$

Goal 2: Adapting to Intraday Auction Price

- The same threshold for every day would give bad results because the price level can be different from day to day (left graph)
- The intraday auction price seems as a good explanatory variable (right graph)

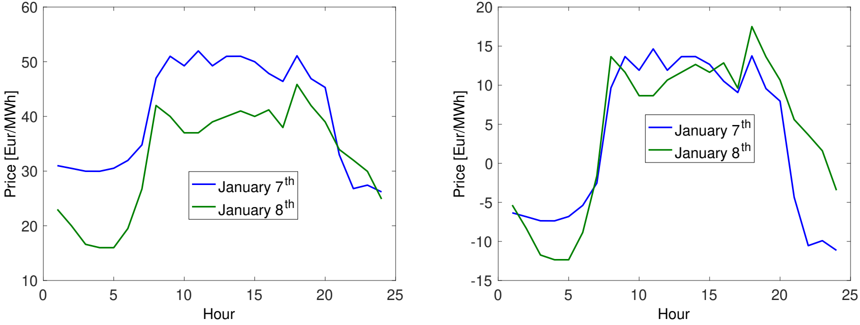


Figure: Price for two different days (left). Price for two different days at which we subtract the average intraday auction price (right).

Goal 2: Adapting to Intraday Auction Price

- The buy threshold is taken as the minimum value of the intraday curve at which we add a security margin that will be learned by the algorithm.
- The sell threshold is taken as the maximum value of the intraday curve at which we subtract a security margin that will be learned by the algorithm.

$$\mu_X \leftarrow p_{\min} + \alpha_1^s (p_{\max} - p_{\min})$$
$$\mu_Y \leftarrow p_{\max} - \alpha_1^b (p_{\max} - p_{\min})$$

Analysis

What we decide

We decide the parametrization of the policy. We start at p_{min} and add a security margin $\alpha_1^s(p_{\text{max}} - p_{\text{min}})$

What the algorithm decide

This security margin α_1^s is learned through experience by the REINFORCE algorithm.

Parameters evolution

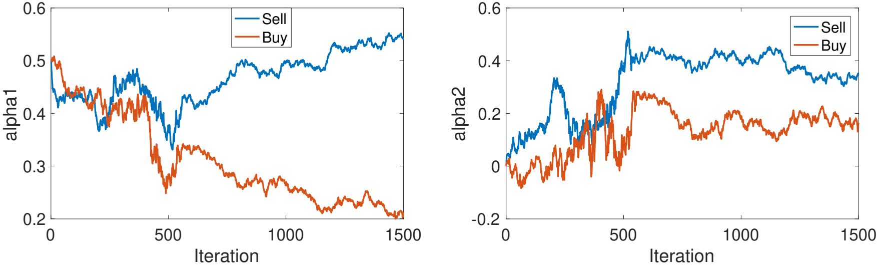


Figure: Evolution of α_1 and α_2.

Parameters evolution

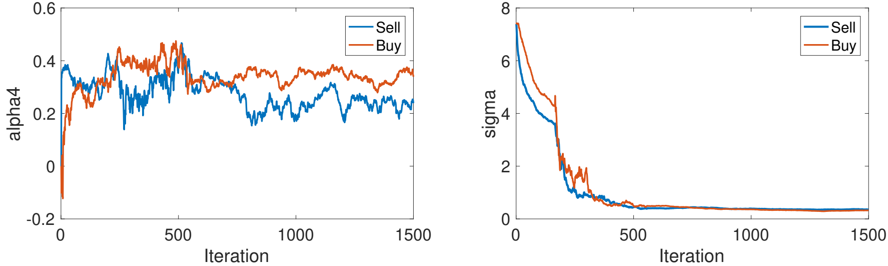


Figure: Evolution of $\alpha_1 4$ and σ.