Efficient Estimation of Upper Bounds on Arbitrage Values for Energy Storage Devices

7th TUD Energy Storage Systems Workshop

Dr Robert Barthorpe, The University of Sheffield

23rd January 2024

Context: Advanced Distributed Storage for grid Benefit

- Funded by Department for Energy Security and Net Zero
- Aim: to develop two domestically focused thermal towards commercialisation
- Phase Change Material (PCM) store: aiming for 14kWh store comprising 3 modules
- Thermochemical storage (TCS): aiming for 144kWh store comprising 3 modules

University of Sheffield
Loughborough University
University of Nottingham UK | CHINA | MALAYSIA
mixergy

ADSorB

Context: Advanced Distributed Storage for grid Benefit

- Funded by Department for Energy Security and Net Zero
- Aim: to develop two domestically focused thermal towards commercialisation
- Phase Change Material (PCM) store: aiming for 14kWh store comprising 3 modules
- Thermochemical storage (TCS): aiming for 144kWh store comprising 3 modules

University of Sheffield

Loughborough University

University of Nottingham
UK | CHINA | MALAYSIA

mixergy

ADSorB

Motivation: Metrics

- Decision making around investment in energy storage assets is a challenging task and is typically reliant upon evaluating the expected performance of the planned asset on the basis of a set of economic metrics.

- The levelised cost of storage (LCOS) has significant limitations: temporal characteristics are not included.

- The net present value (NPV) describes the discounted present value of an asset considering both costs and revenue streams. As such, it offers a means of incorporating temporal behaviour including arbitrage value into the value calculation.

Levelised Cost of Storage (LCOS)

$$LCOS\left[\frac{£}{kWh_{out}}\right] = \frac{CAPEX + \sum_{n=1}^{N}\frac{OPEX_n}{(1+r)^n} + \sum_{n=1}^{N}\frac{Charge\ cost_n}{(1+r)^n} + \frac{EOL}{(1+r)^{N+1}}}{\sum_{n=1}^{N}\frac{Energy\ out_n}{(1+r)^n}}$$

Net Present Value (NPV)

$$NPV[£] = \sum_{n=1}^{N}\frac{Revenue_n}{(1+r)^n} - CAPEX - \sum_{n=1}^{N}\frac{OPEX_n}{(1+r)^n} - \frac{EOL}{(1+r)^{N+1}}$$

ADSorB

Motivation: Metrics

- Decision making around investment in energy storage assets is a challenging task and is typically reliant upon evaluating the expected performance of the planned asset on the basis of a set of economic metrics.

- The levelised cost of storage (LCOS) has significant limitations: temporal characteristics are not included.

- The net present value (NPV) describes the discounted present value of an asset considering both costs and revenue streams. As such, it offers a means of incorporating temporal behaviour including arbitrage value into the value calculation.

Levelised Cost of Storage (LCOS)

$$LCOS\left[\frac{\pounds}{kWh_{out}}\right] = \frac{CAPEX + \sum_{n=1}^{N}\frac{OPEX_n}{(1+r)^n} + \sum_{n=1}^{N}\frac{Charge\ cost_n}{(1+r)^n} + \frac{EOL}{(1+r)^{N+1}}}{\sum_{n=1}^{N}\frac{Energy\ out_n}{(1+r)^n}}$$

Net Present Value (NPV)

$$NPV[\pounds] = \sum_{n=1}^{N}\frac{Revenue_n}{(1+r)^n} - CAPEX - \sum_{n=1}^{N}\frac{OPEX_n}{(1+r)^n} - \frac{EOL}{(1+r)^{N+1}}$$

Question: How to efficiently estimate '$Revenue_n$'?

ADSorB

Methodology: Overview

- This study focuses on the development of an efficient method for estimating an upper bound on the arbitrage value that may be achieved by a given storage device, such that it may be included with the NPV metric.
- The approach employs graphical modelling and dynamic programming techniques.
- Outcomes are illustrated for a selection of archetypal storage devices.

Ingesting and pre-processing historical (or forecast) time series data on available energy prices and associated carbon intensities.

↓

Identifying peaks and troughs in the time series that represent candidate ‘sell’ and ‘buy’ points, respectively.

↓

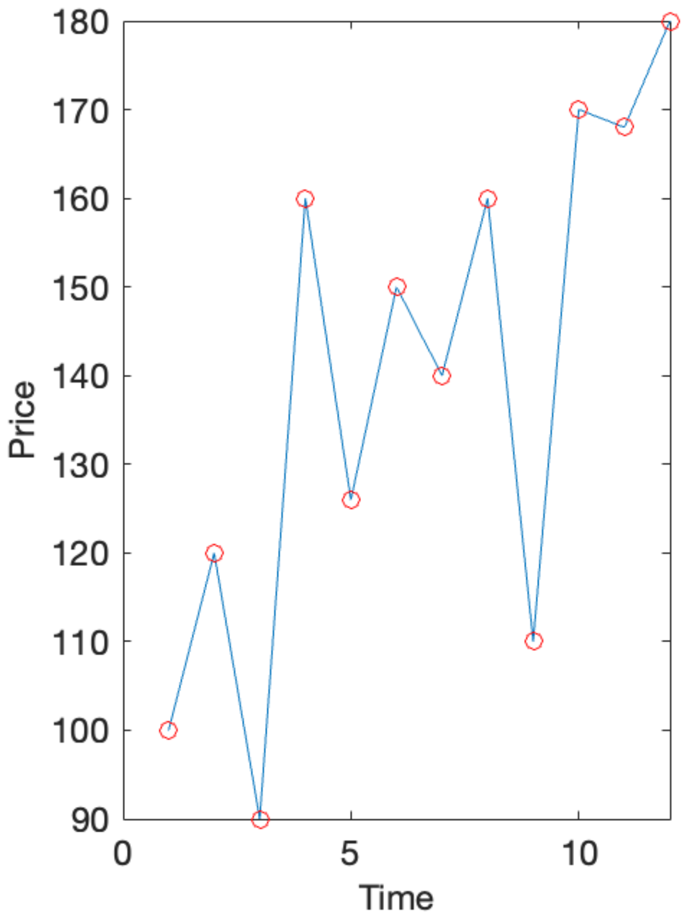
Setting up a directed acyclic graph (DAG) to represent the buy/sell/hold actions (‘edges’) connecting the identified buy/sell points (‘nodes’).

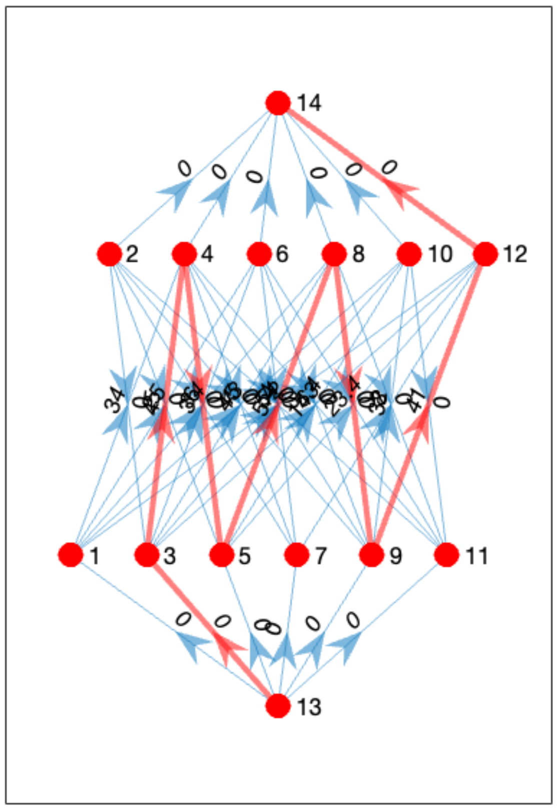
↓

Searching for path through this graph that optimises carbon/cost benefits, efficiently returning an estimated upper bound on cost and/or carbon savings.

Methodology: Stock trading

- Aim: identify the maximum ‘profit’ that may be realised over a period of time through optimising trading decisions
- An *ex-post* analysis – assumes data perfectly known
- Essentially “*buy low, sell high*”
- Transaction costs may be included for both buy and sell actions
- Optimisation problem may be efficiently solved by:
    1. Identifying trading points in price history
    2. Using these to build a bipartite directed acyclic graph (DAG)
    3. Finding the most profitable path through the DAG
- Example shown is for transaction cost c=10%





Brandouy O, Mathieu P, Veryzhenko I, Algorithmic determination of the maximum possible earnings for investment strategies, Decision Support Systems, Volume 54, Issue 2 (2013) 816-82

Methodology: Stock trading

Graph construction

1. Trading edge: An edge with associated weight $R_{i,j}$ is constructed between a candidate buy point i and candidate sell point j if and only if the trade is:
   - Temporally consistent
   - Profitable

$$t_j > t_i$$
$$R_{i,j} = p_j(1 - c) - p_i(1 + c) \geq 0$$

2. Forward edge: A zero-weighted edge is constructed between candidate sell point m and candidate buy point n if and only if the trade is:
   - Temporally consistent

$$t_n > t_m$$

Solution

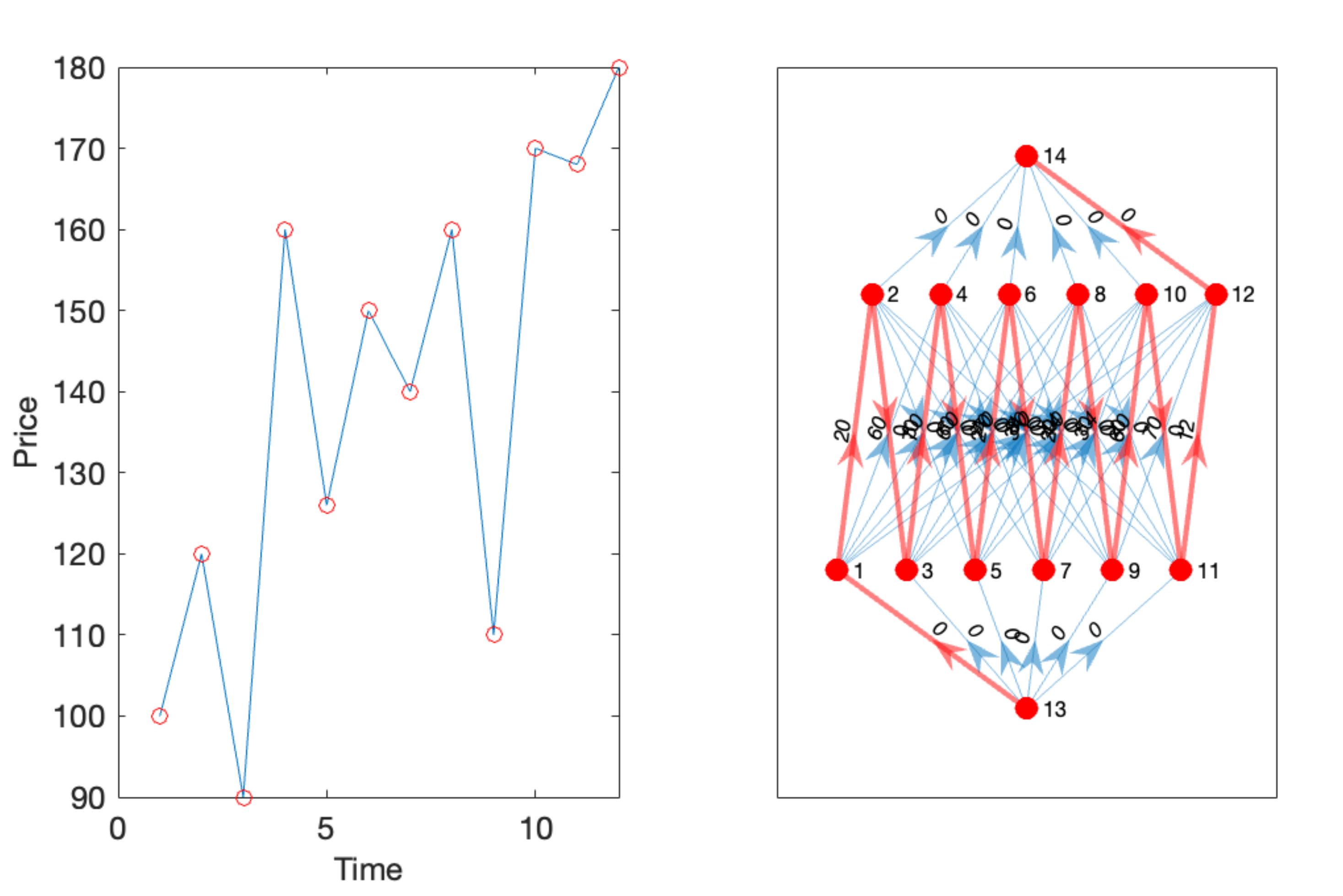
- Find longest (i.e. most profitable) path through the directed acyclic graph, G
  - Or equivalently the shortest path through the negative graph, $-G$
- Solves in linear time $\mathcal{O}(n)$

With:
- Transaction cost, c

ADSorB

Methodology: Stock trading

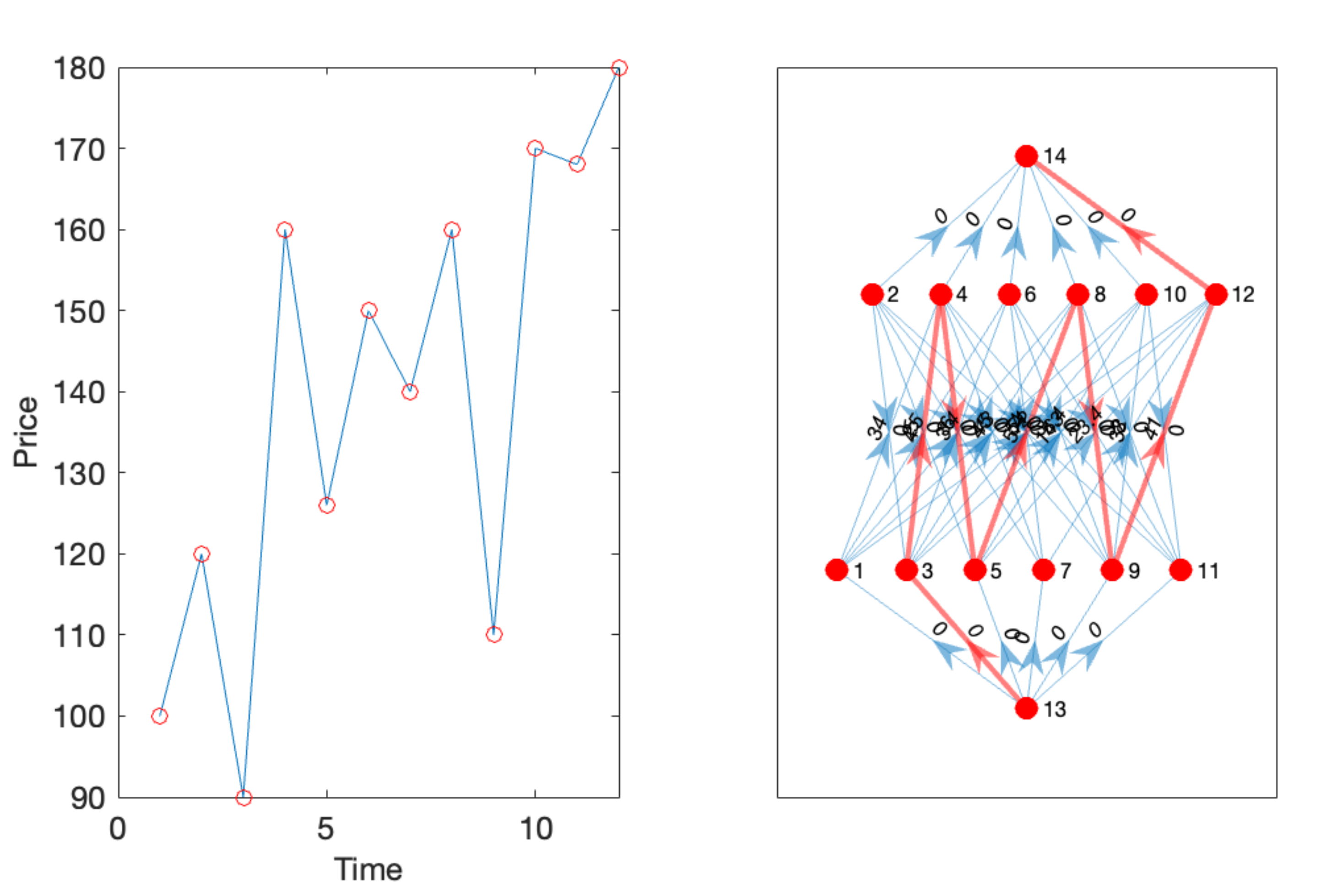
- Aim: identify the maximum ‘profit’ that may be realised over a period of time through optimising trading decisions
- An *ex-post* analysis – assumes data perfectly known
- Essentially “*buy low, sell high*”
- Transaction costs may be included for both buy and sell actions
- Optimisation problem may be efficiently solved by:
    1. Identifying trading points in price history
    2. Using these to build a bipartite directed acyclic graph (DAG)
    3. Finding the most profitable path through the DAG
- Example shown is for transaction cost c=0%



ADSorB

Methodology: Stock trading

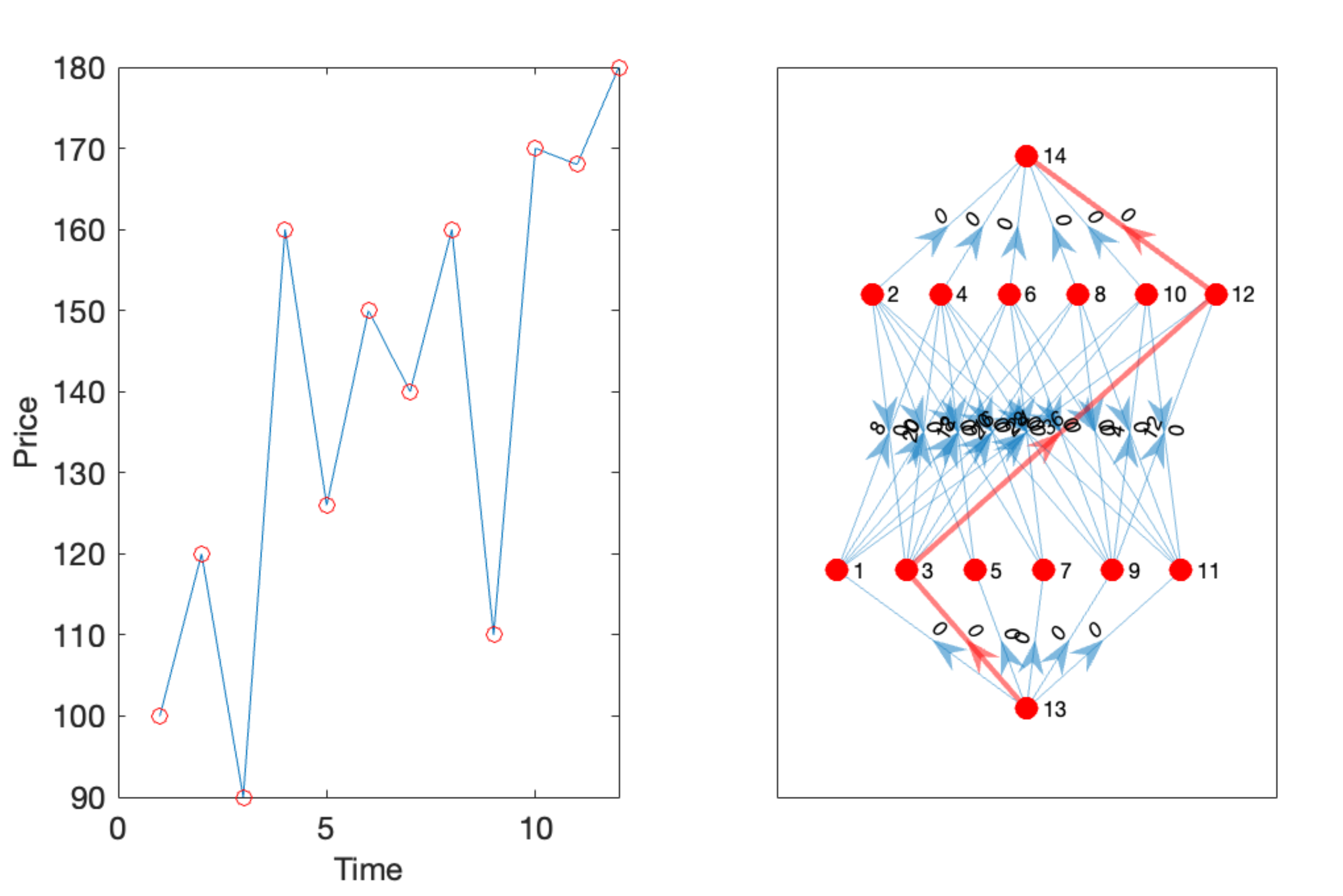
- Aim: identify the maximum 'profit' that may be realised over a period of time through optimising trading decisions
- An *ex-post* analysis – assumes data perfectly known
- Essentially "*buy low, sell high*"
- Transaction costs may be included for both buy and sell actions
- Optimisation problem may be efficiently solved by:
    1. Identifying trading points in price history
    2. Using these to build a bipartite directed acyclic graph (DAG)
    3. Finding the most profitable path through the DAG
- Example shown is for transaction cost c=10%



ADSorB

Methodology: Stock trading

- Aim: identify the maximum ‘profit’ that may be realised over a period of time through optimising trading decisions
- An *ex-post* analysis – assumes data perfectly known
- Essentially “*buy low, sell high*”
- Transaction costs may be included for both buy and sell actions
- Optimisation problem may be efficiently solved by:
    1. Identifying trading points in price history
    2. Using these to build a bipartite directed acyclic graph (DAG)
    3. Finding the most profitable path through the DAG
- Example shown is for transaction cost c=20%



ADSorB

Methodology: Stock trading

Graph construction

1. Trading edge: An edge with associated weight $R_{i,j}$ is constructed between a candidate buy point i and candidate sell point j if and only if the trade is:
   - Temporally consistent
   - Profitable

2. Forward edge: A zero-weighted edge is constructed between candidate sell point m and candidate buy point n if and only if the trade is:
   - Temporally consistent

$$t_j > t_i$$
$$R_{i,j} = p_j(1-c) - p_i(1+c) \geq 0$$

$$t_n > t_m$$

With:
- Transaction cost, c

Solution

- Find longest (i.e. most profitable) path through the directed acyclic graph, G
  - Or equivalently the shortest path through the negative graph, $-G$
- Solves in linear time $\mathcal{O}(n)$

ADSorB

Methodology: Application to Arbitrage

Graph construction

1. Trading edge: An edge with associated weight $R_{i,j}$ is constructed between a candidate buy point i and candidate sell point j if and only if the trade is:
   - Temporally consistent
   - Profitable

$$t_j > t_i$$
$$R_{i,j} = p_j \eta_{dch} f(t_j, t_i) - p_i / \eta_{ch} \geq 0$$

2. Forward edge: A zero-weighted edge is constructed between candidate sell point m and candidate buy point n if and only if the trade is:
   - Temporally consistent

$$t_n > t_m$$

Solution

- Find longest (i.e. most profitable) path through the directed acyclic graph, G
  - Or equivalently the shortest path through the negative graph, $-G$
- Solves in linear time $\mathcal{O}(n)$

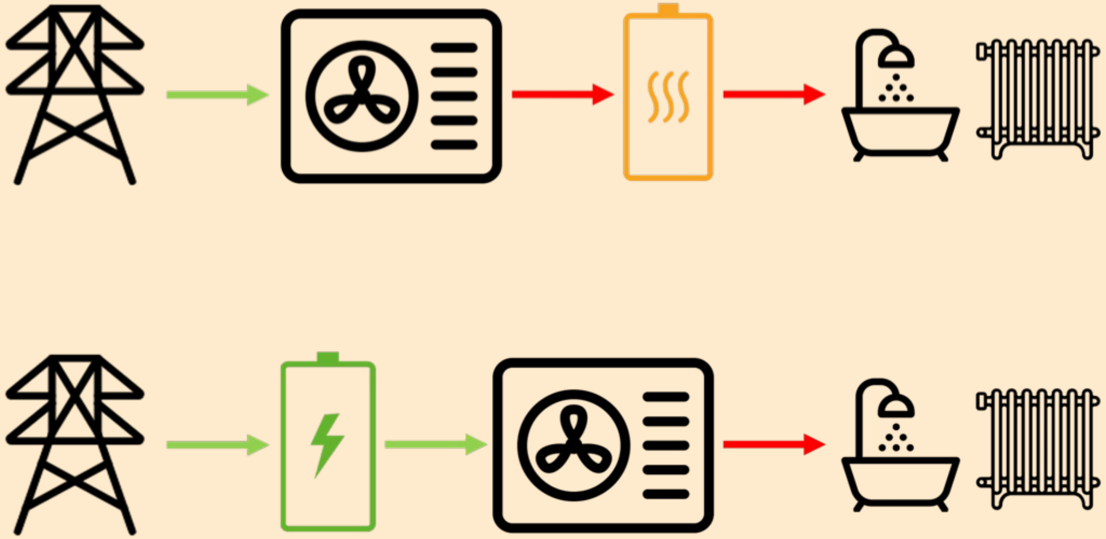
With:
- Charge efficiency η_{ch}
- Discharge efficiency η_{dch}
- Standing loss function, $f(t_j, t_i)$

ADSorB

Methodology: Application to Arbitrage

“Effective” price and carbon records

- Incorporating an air source heat pump model (Staffell et al, 2012) and local weather data
- Assumes either electrical power converted to thermal energy via HP and stored as heat, or electrical power stored and converted to heat via HP as required
- Buy/sell price accounts for changing TOUT prices within the charge/discharge period, and could be straightforwardly extended to account for non-constant/linear charge rates
- Peak picking applied to identify candidate trading points, promoting sparsity



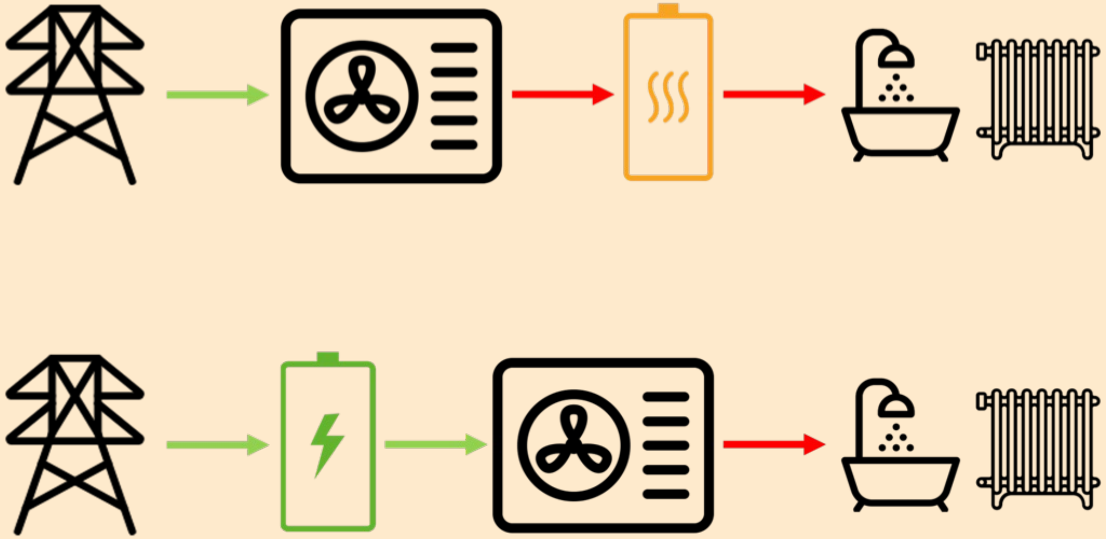
Staffell I, Brett D, Brandon N, Hawkes A, A review of domestic heat pumps, *Energy & Environmental Science*, Volume 5, (2012) 9291-9306

ADSorB

Methodology: Application to Arbitrage

"Effective" price and carbon records

- Incorporating an air source heat pump model (Staffell et al, 2012) and local weather data
- Assumes either electrical power converted to thermal energy via HP and stored as heat, or electrical power stored and converted to heat via HP as required
- Buy/sell price accounts for changing TOUT prices within the charge/discharge period, and could be straightforwardly extended to account for non-constant/linear charge rates
- Peak picking applied to identify candidate trading points, promoting sparsity



Staffell I, Brett D, Brandon N, Hawkes A, A review of domestic heat pumps, *Energy & Environmental Science*, Volume 5, (2012) 9291-9306

Demonstration: Price Arbitrage

- Aim: identify the maximum cost reduction that may be realised over a period of time through optimising charge/discharge actions

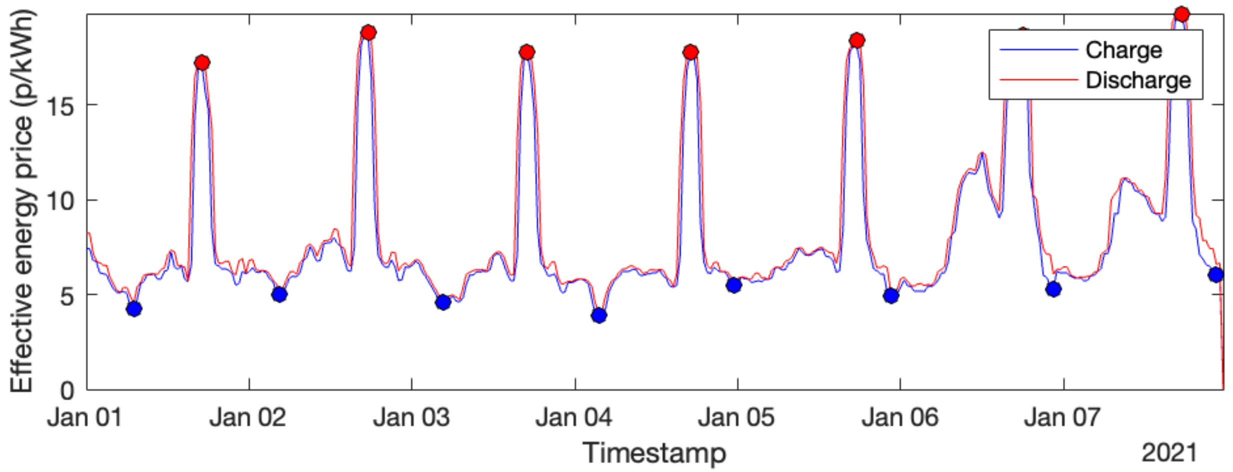
- "Price" histories:

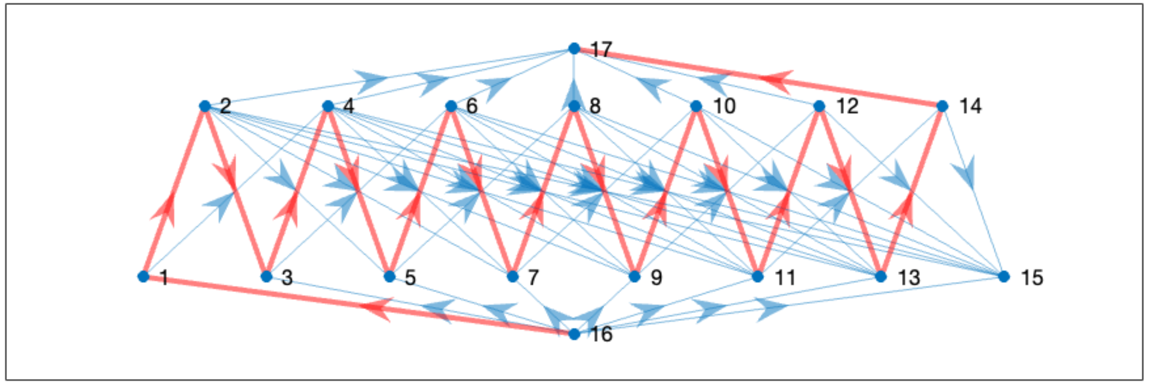
$$\text{Effective Price} = \frac{\text{Import Price}}{\text{COP}}$$

- Data for 2021, using:
  - Octopus Agile Import TOUT
  - Weather for Nottingham, UK

- For the example shown:
  - Capacity = 10 kWh
  - Charge rate = 8 kW
  - Discharge rate = 8 kW
  - Charge efficiency = 93%
  - Discharge efficiency = 93%
  - Standing losses = 0.42%/HH period
  - Min peak separation = 12 hours

- Results: Upper bound of £0.68/kWh cost reduction, resulting from 7 charge cycles during period





ADSorB

Demonstration: Price Arbitrage

- Aim: identify the maximum cost reduction that may be realised over a period of time through optimising charge/discharge actions

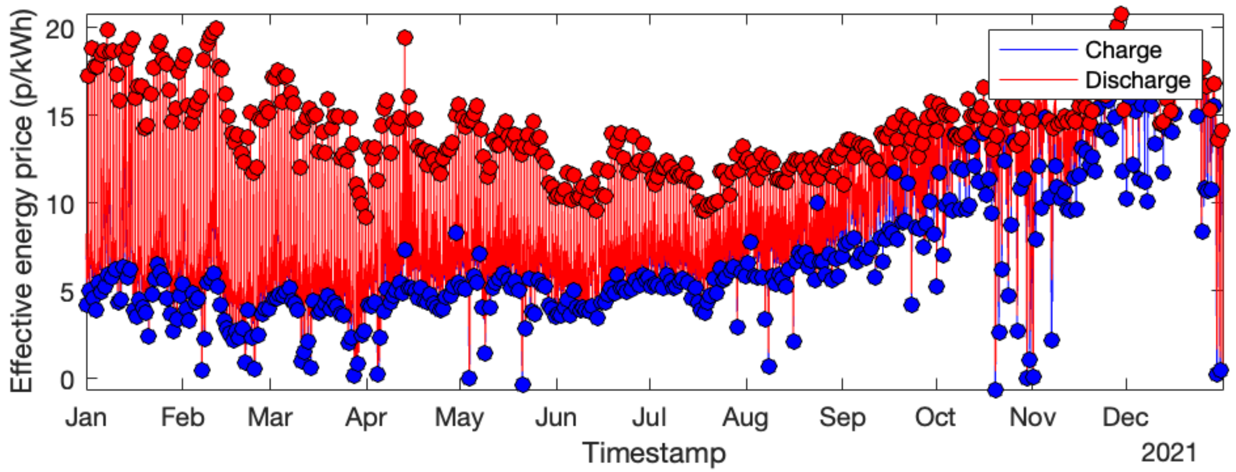
- "Price" histories:

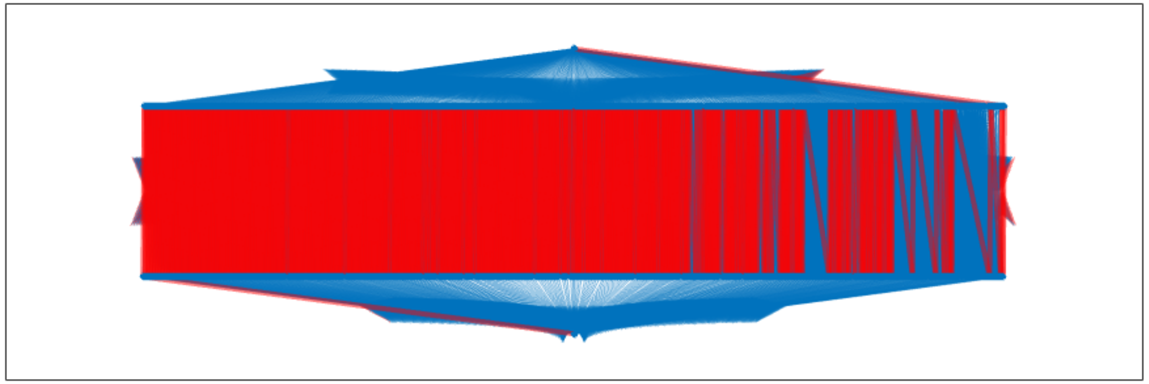
$$\text{Effective Price} = \frac{\text{Import Price}}{\text{COP}}$$

- Data for 2021, using:
  - Octopus Agile Import TOUT
  - Weather for Nottingham, UK

- For the example shown:
  - Capacity = 10 kWh
  - Charge rate = 8 kW
  - Discharge rate = 8 kW
  - Charge efficiency = 93%
  - Discharge efficiency = 93%
  - Standing losses = 0.42%/HH period
  - Min peak separation = 12 hours

- Results: Upper bound of £17.81/kWh cost reduction, resulting from 297 charge cycles during period





ADSorB

Demonstration: Carbon Arbitrage

- Aim: identify the maximum cost reduction that may be realised over a period of time through optimising charge/discharge actions

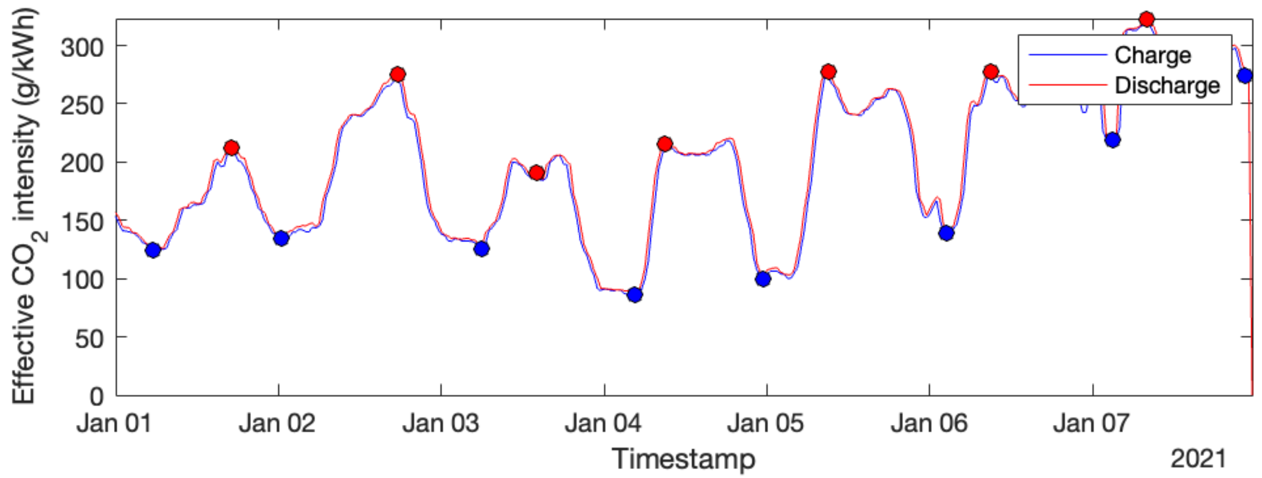
- "Price" histories:

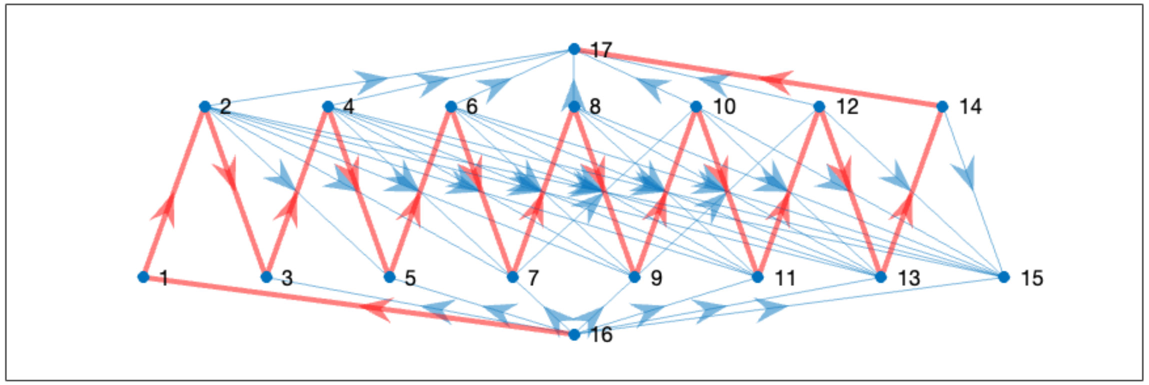
$$\text{Effective } CO_2 \text{ Intensity} = \frac{GCI}{COP}$$

- Data for 2021, using:
  - National Grid Carbon Intensity (GCI)
  - Weather for Nottingham, UK

- For the example shown:
  - Capacity = 10 kWh
  - Charge rate = 8 kW
  - Discharge rate = 8 kW
  - Charge efficiency = 93%
  - Discharge efficiency = 93%
  - Standing losses = 0.42%/HH period
  - Min peak separation = 12 hours

- Results: Upper bound of 0.5kgCO2/kWh of storage, resulting from 7 charge cycles during period





ADSorB

Demonstration: Carbon Arbitrage

- Aim: identify the maximum cost reduction that may be realised over a period of time through optimising charge/discharge actions

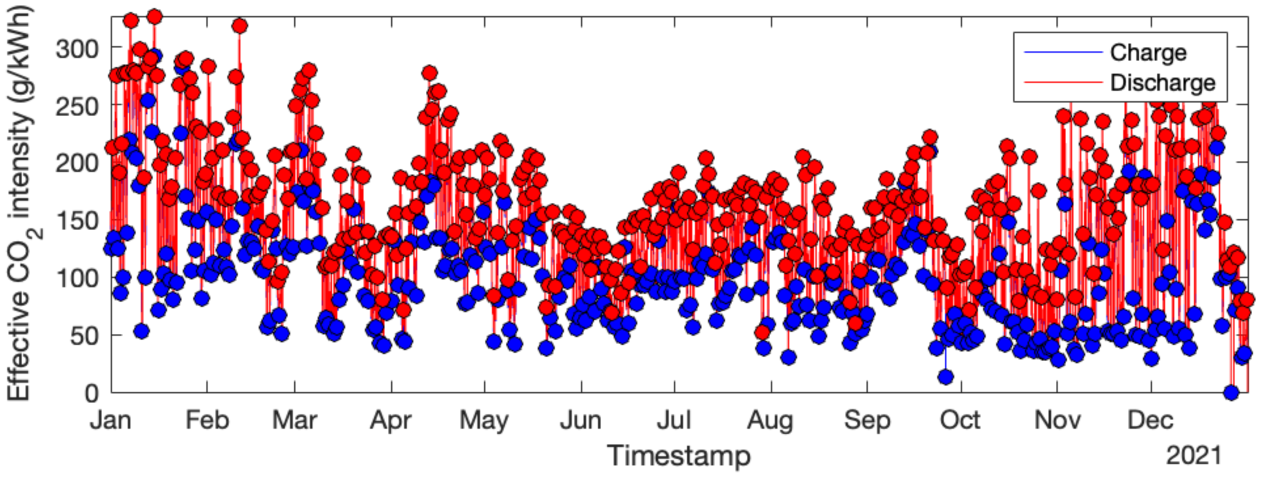
- "Price" histories:

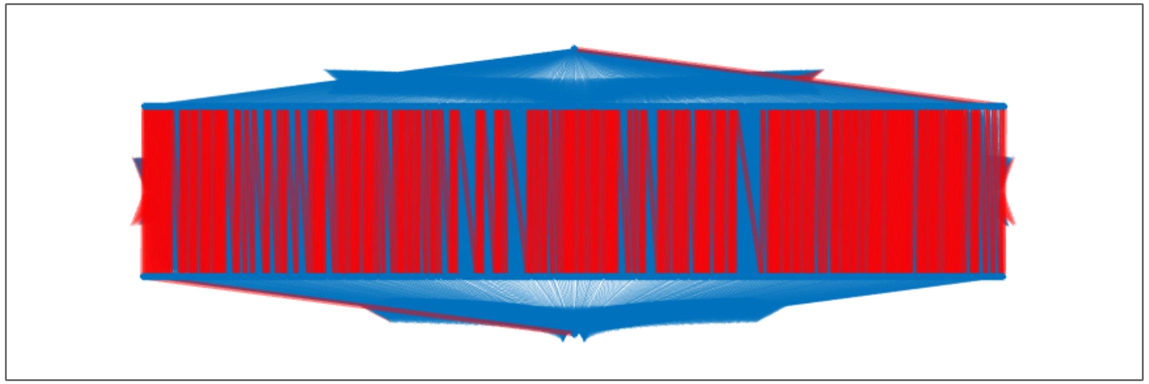
$$\text{Effective } CO_2 \text{ Intensity} = \frac{GCI}{COP}$$

- Data for 2021, using:
  - National Grid Carbon Intensity (GCI)
  - Weather for Nottingham, UK

- For the example shown:
  - Capacity = 10 kWh
  - Charge rate = 8 kW
  - Discharge rate = 8 kW
  - Charge efficiency = 93%
  - Discharge efficiency = 93%
  - Standing losses = 0.42%/HH period
  - Min peak separation = 12 hours

- Results: Upper bound of 13.0kgCO2/kWh of storage, resulting from 283 charge cycles during period





ADSorB

Extension: Limited Charge Cycles

Floyd-Warshall Algorithm

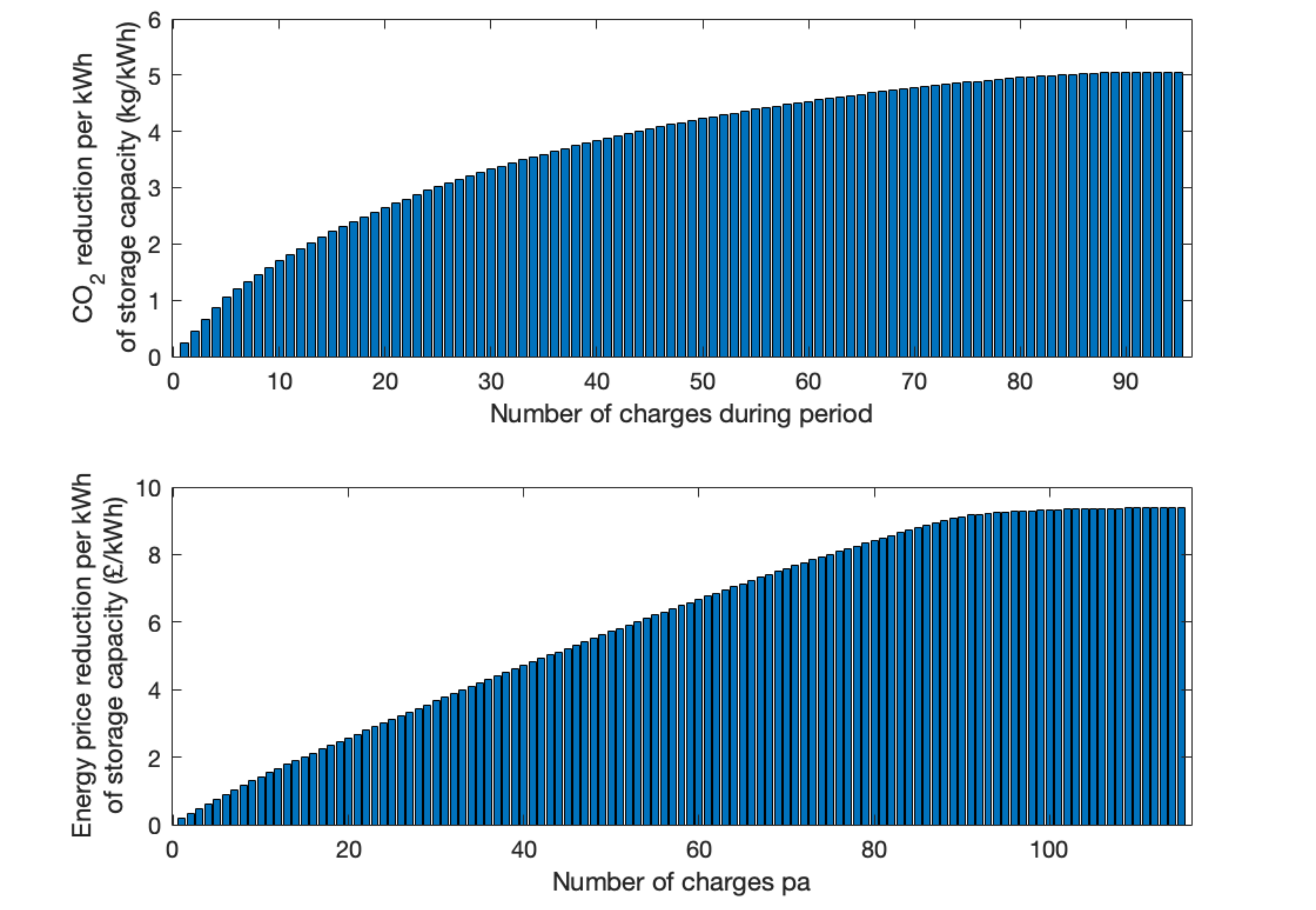
Returns the shortest path between pairs of vertices for a weighted graph $G(V, E)$

for each $d \in V$
    $dist[d][d] \leftarrow 0$
end

for each edge $(s, t) \in E$
    $dist[s][t] \leftarrow weights\ (s, t)$
end

$n = number\ of\ edges\ in\ G = |E|$
for $k = 1$ **to** n
    for $i = 1$ **to** n
        for $j = 1$ **to** n
            if $dist[i, j] > dist[i, k] + dist[k, j]$
            then $dist[i, j] \leftarrow dist[0, k] + dist[k, j]$
        end
    end
end

Running time of $\mathcal{O}(n^3)$
n.B if no edge (s, t) exists, $dist[s][t]$ set to ∞



ADSorB

R. Floyd, Algorithm 97, shortest path algorithms, *Operations Research*, Volume 17 (1969) 395–412

Extension: Limited Charge Cycles

Floyd-Warshall Algorithm

Returns the shortest path between pairs of vertices for a directed weighted graph $G(V, E)$

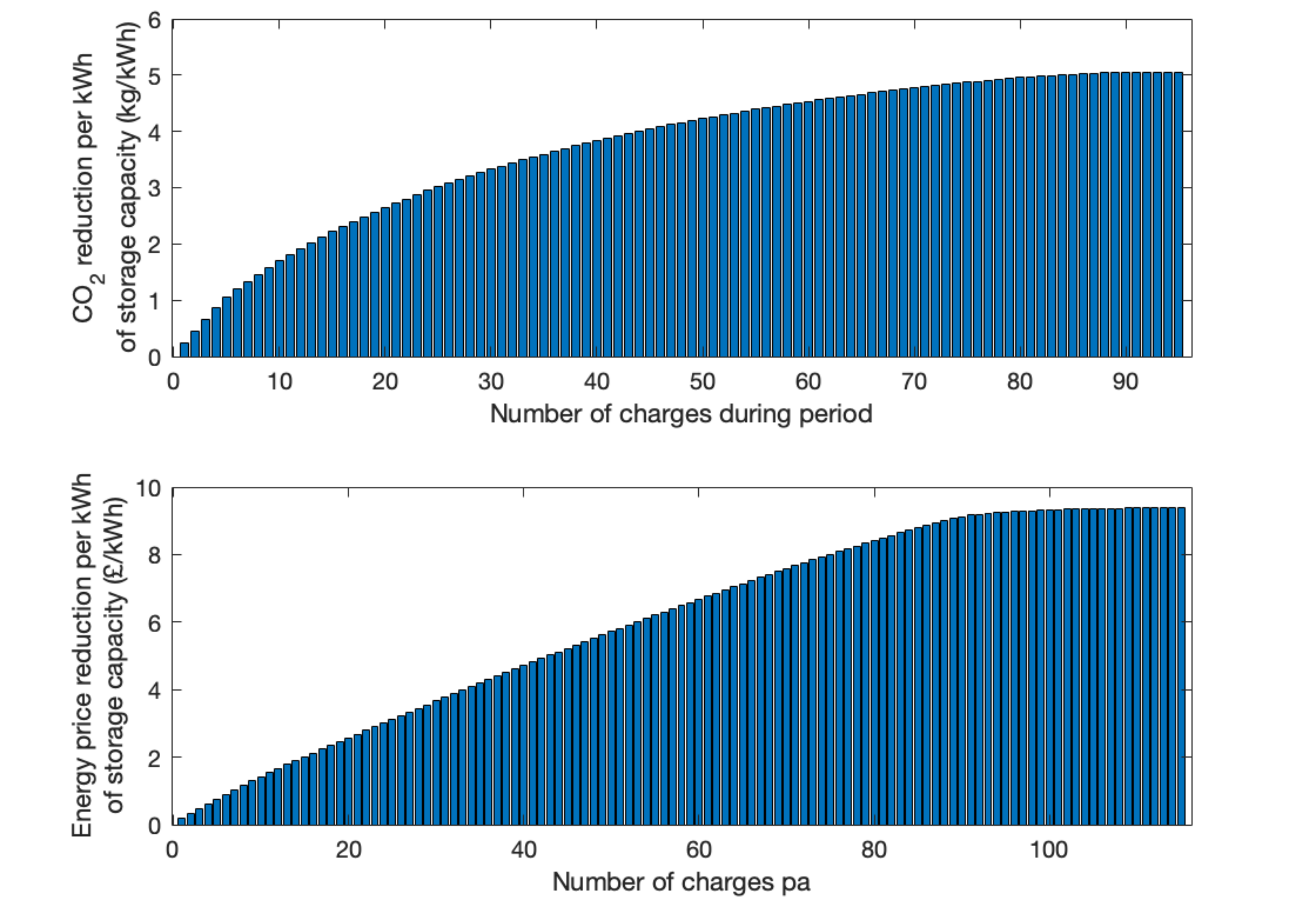
```
for each d ∈ V
    dist[d][d] ← 0
end

for each edge (s,t) ∈ E
    dist[s][t] ← weights (s,t)
end

n = number of edges in G = |E|
for k = 1 to n
    for j = k to n
        if dist[0,j] > dist[0,k] + dist[k,j]
        then dist[0,j] ← dist[0,k] + dist[k,j]
    end
end
```

Running time of $\mathcal{O}(n^2)$

n.B if no edge (s,t) exists, $dist[s][t]$ set to ∞



ADSorB

R. Floyd, Algorithm 97, shortest path algorithms, *Operations Research*, Volume 17 (1969) 395–412

Demonstration: Price Arbitrage

Store	Capacity. [kWh]	Charge Power [kW]	Discharge Power [kW]	CAPEX [£]	OPEX/pa [£]	EoL [£]	Charge Eff [%]	Discharge Eff [%]	Standing Losses [W]	Life [years]
Elec 1	10.4	3.6	5	10000	50	100	95	95	4	10
Therm 1	7	15	15	1000	50	100	100	100	77	25
Therm 2	10	15	15	1250	50	100	100	100	32	25
Therm 3	144	15	15	1500	50	100	80	100	0	25

Levelised Cost of Storage (LCOS)

Store	Cycles pa	Charge efficiency [%]	Discharge efficiency [%]	LCOS [p/kWh_{sto}]
Elec 1	365	95	285	16.9
Therm 1	365	300	100	9.3
Therm 2	365	300	100	8.4
Therm 3	52	240	100	8.1

Net Present Value (NPV)

Store	Cycles pa	Arbitrage value pa		NPV [£]
		/kWh_{cap} [£]	Total [£]	
Elec 1	338	64.83	674	-4521
Therm 1	341	24.45	171	2362
Therm 2	340	27.77	278	4776
Therm 3	52	1.99	286	4747

ADSorB

Advantages, Limitations and Future Work

Advantages

- Provides an efficient ($\mathcal{O}(n)$) means of evaluating an upper bound on arbitrage value for a candidate system
- Presents possibility of including nonlinear charge/discharge behaviour without increasing computation cost
- Provides a means of evaluating performance with a reduced number of charge/discharge cycles

Limitations

- Represents an approximation
    1. "All in" strategy – assumes demand exists; omits consideration of part-charging
    2. Peak picking to limit graph size
- *Ex-post* analysis; data must be available. Achieving performance in practice is the remit of effective control.
- Evaluation for limited charge cycles (Floyd) is far slower than for overall optimum (DAG shortest path)

Future work

- Full comparison to alternative approaches, principally linear programming (LP)
- Extension to multi-objective (cost and carbon) optimisation
- Development of a sharable software application

ADSorB

Any questions?

Dr Robert Barthorpe, The University of Sheffield
r.j.barthorpe@sheffield.ac.uk
www.adsorb.ac.uk