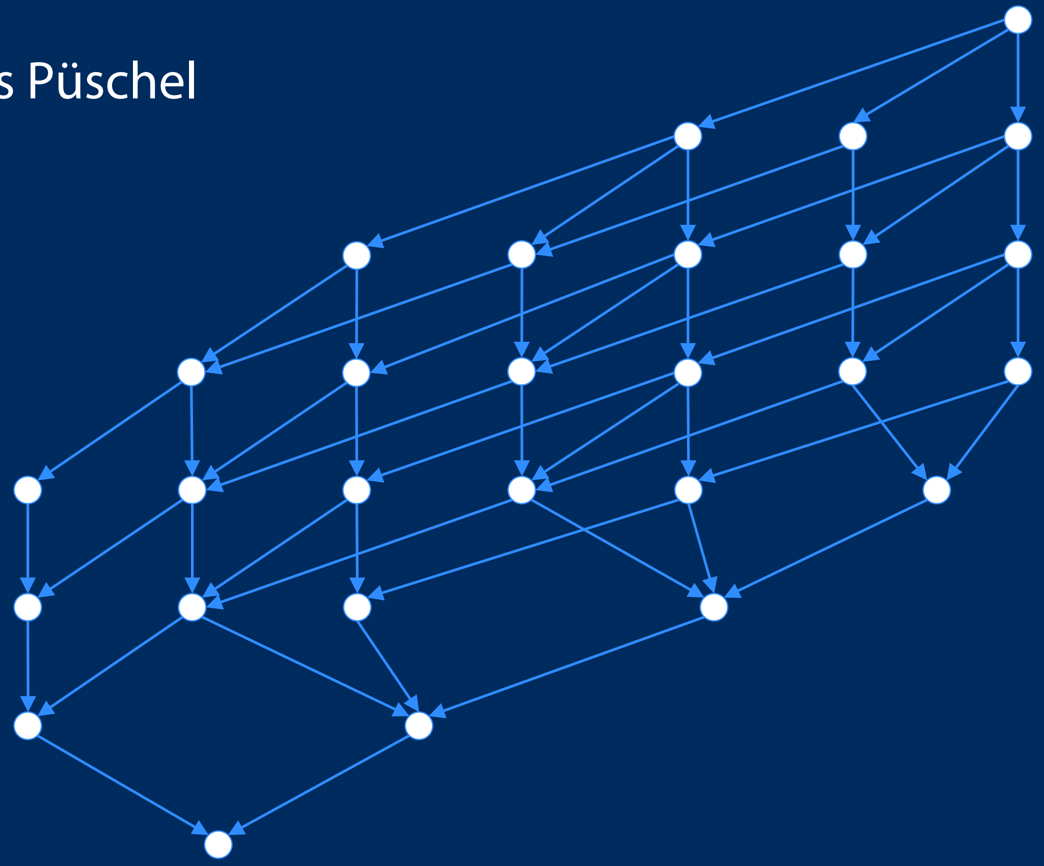
Wiener Filter on Meet/Join Lattices

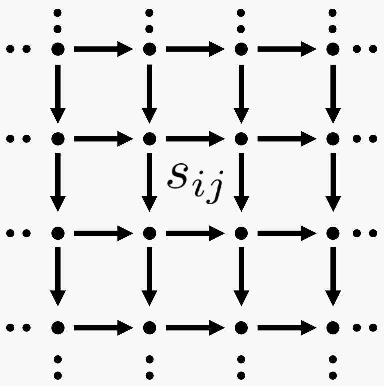
Bastian Seifert and Chris Wendler and Markus Püschel

IEEE ICASSP 2021



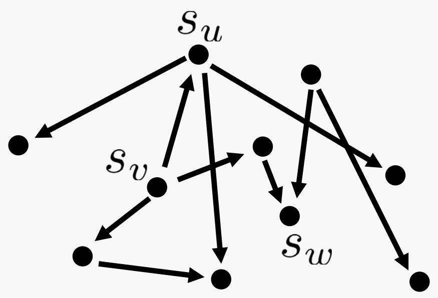
Computer Science
ETH zürich

Classical DSP



Signals indexed by time/space

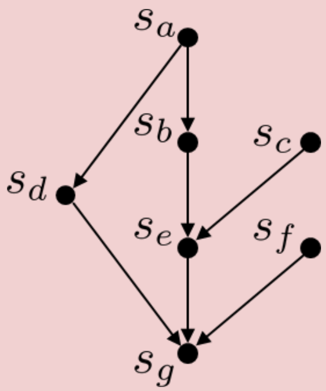
Graph DSP



Signals indexed by nodes of a graph

Shumann 2012
Sandryhaila 2013

New Discrete Lattice SP



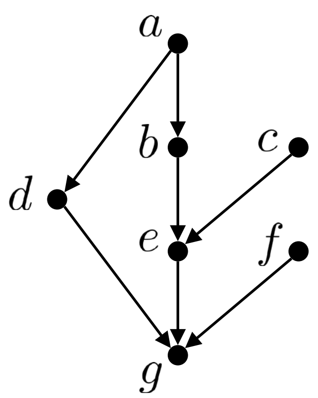
Signals indexed by a meet/join lattice
Instantiation of algebraic signal processing theory

ICASSP 2019

Goal shift, convolution/filtering, Fourier transform, frequency response, sampling, **for lattice signals**

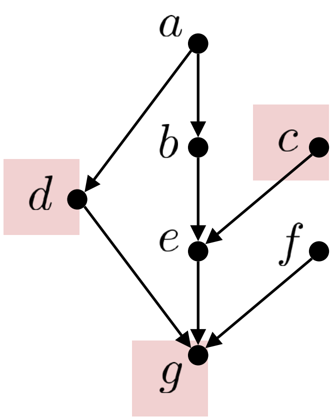
Meet Semilattice

Finite set L with **partial order** $\leq$ and
meet operation $x \wedge y$ (greatest lower bound)



Meet Semilattice

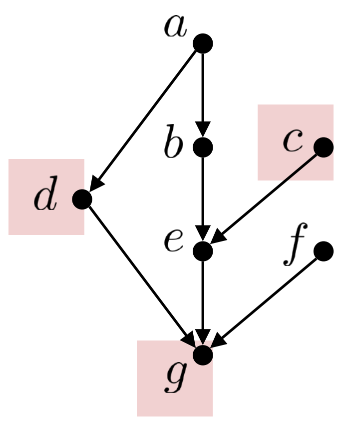
Finite set L with **partial order** $\leq$ and
meet operation $x \wedge y$ (greatest lower bound)



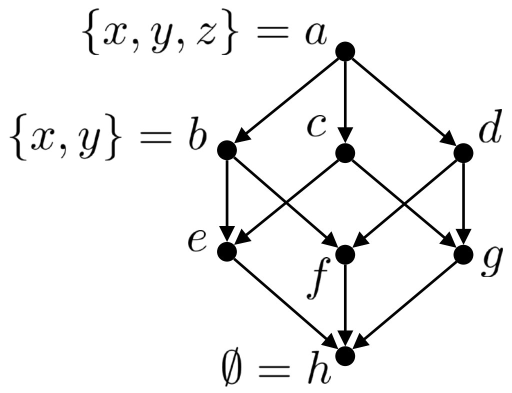
For example, $c \wedge d = g$

Meet Semilattice

Finite set L with **partial order** $\leq$ and
meet operation $x \wedge y$ (greatest lower bound)



For example, $c \wedge d = g$



Powerset Lattice

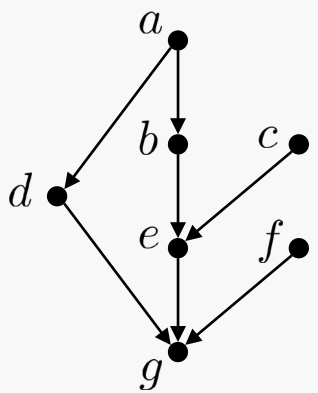
Partial order $\subseteq$, meet $\cap$

Join Semilattice

Analogous

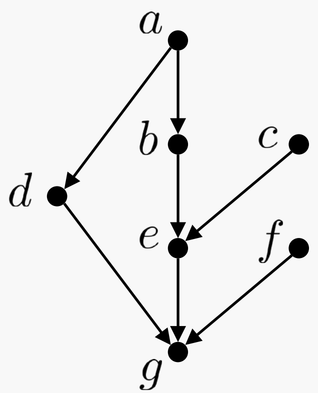
Discrete Lattice Signal Processing

Lattice

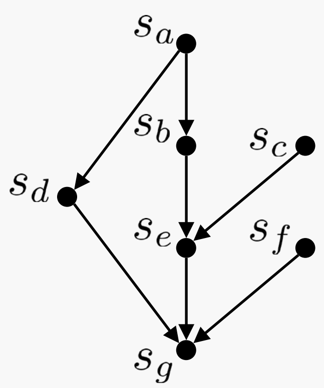


Discrete Lattice Signal Processing

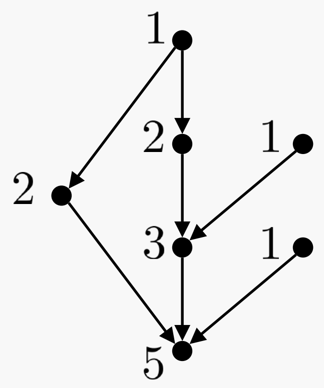
Lattice



Signal $\mathbf{s} = (s_x)_{x \in L} \in \mathbb{R}^n$

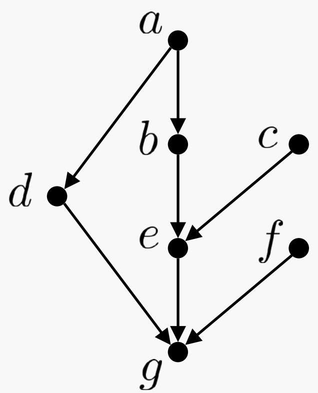


Concrete Example



Discrete Lattice Signal Processing

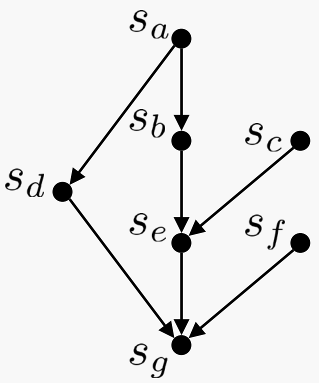
Lattice



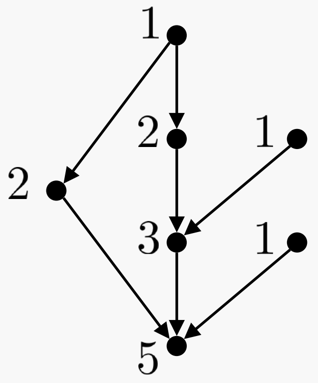
Shift(s) by $q \in L$

$$T_q \mathbf{s} = (s_{x \wedge q})_{x \in L}$$

Signal $\mathbf{s} = (s_x)_{x \in L} \in \mathbb{R}^n$

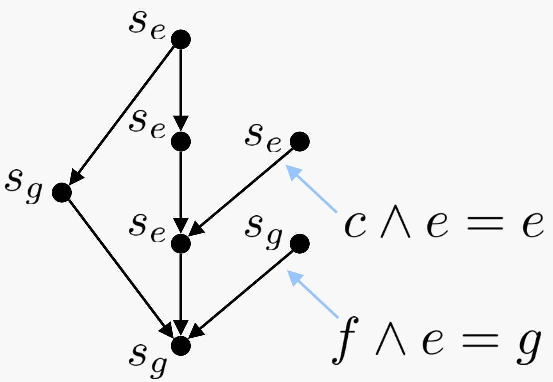


Concrete Example

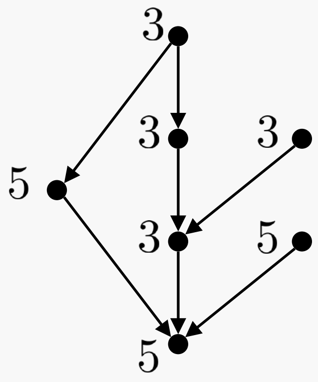


$T_e \mathbf{s}$

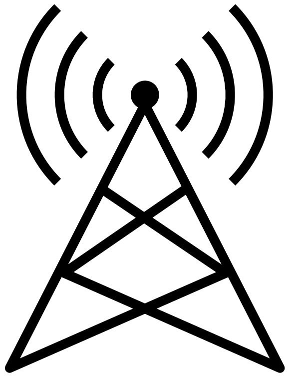
Shifted Signal (by e)



Shifted Example (by e)

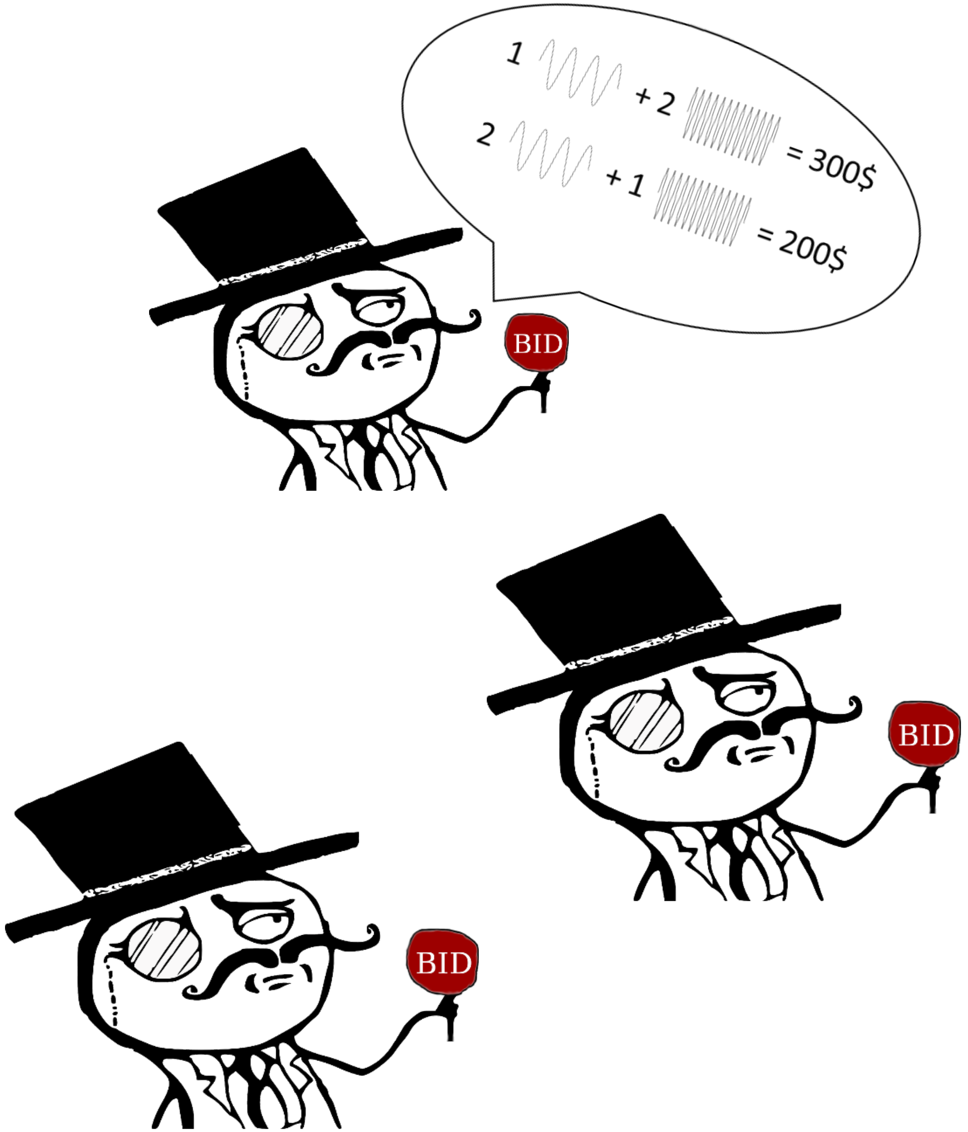


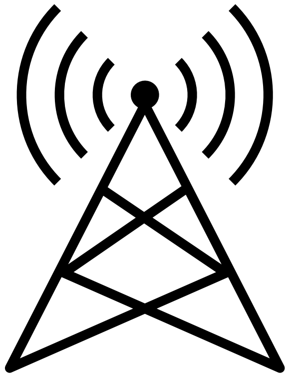
Spectrum Auctions



2 Licenses x
1 License y

Spectrum Auctions



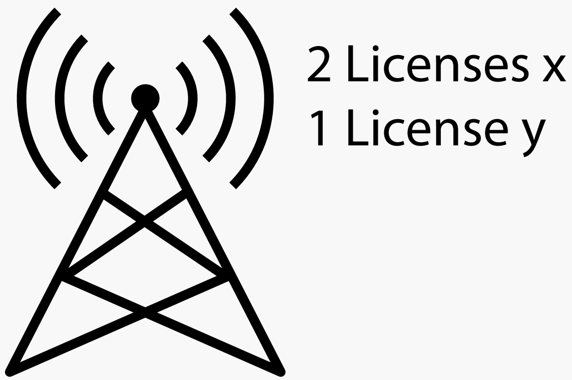


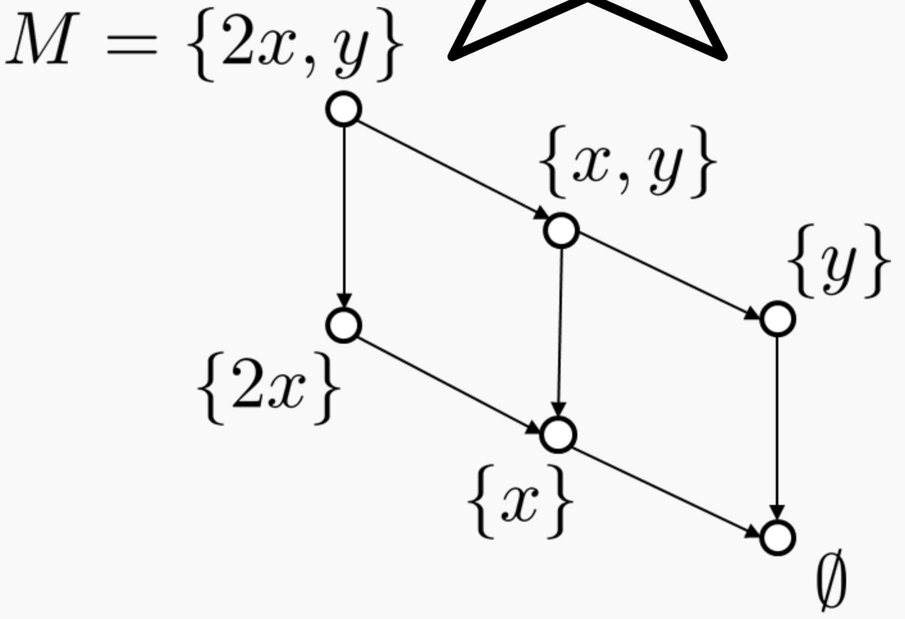
2 Licenses x
1 License y

Spectrum Auctions

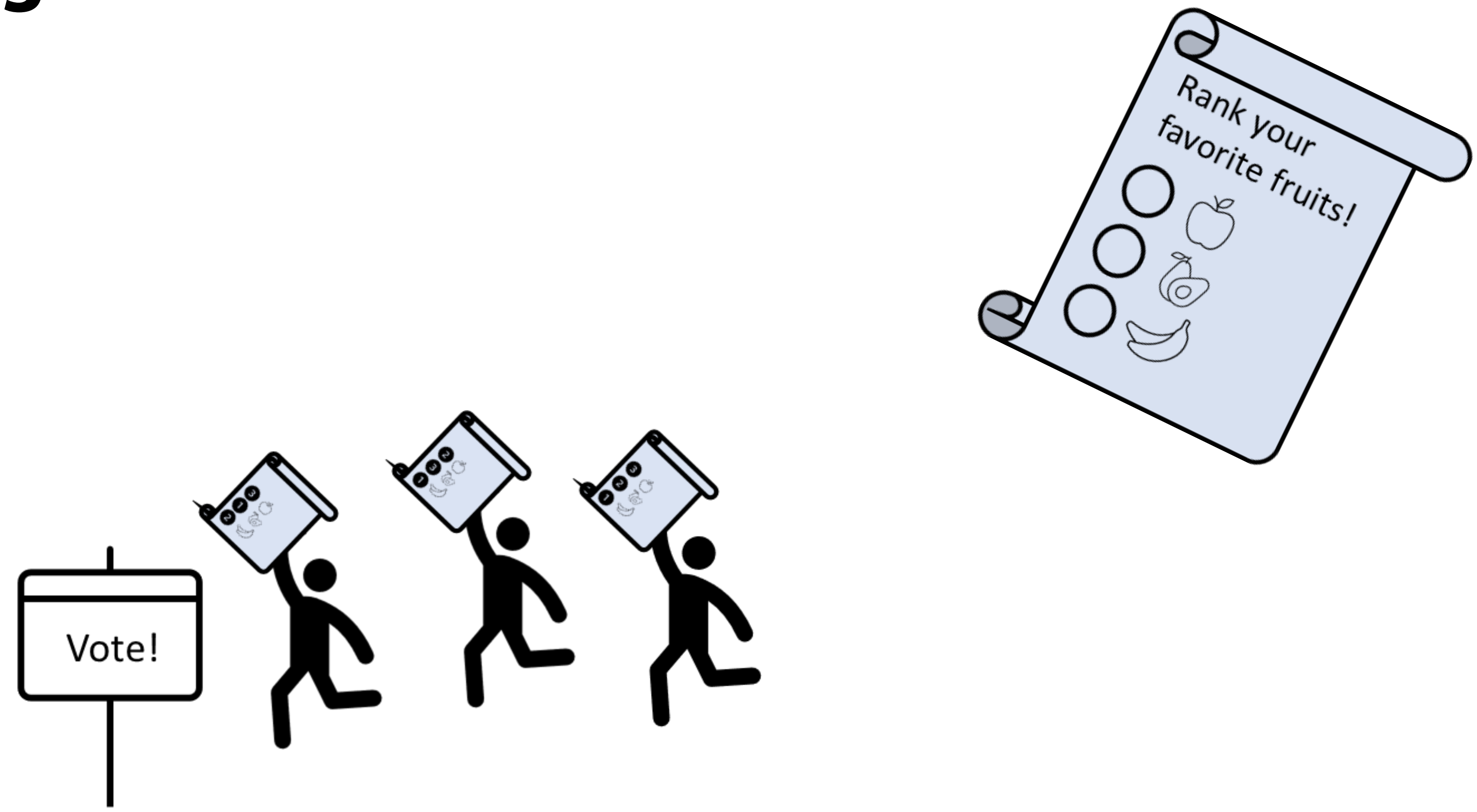
Multiset lattice
Elements: *Submultisets of multiset M*
Partial Order: $\subseteq$
Meet: $\cap$

Multiset of licenses = lattice
Bidder = signal of values for each submultiset of licenses





Rankings



Rankings

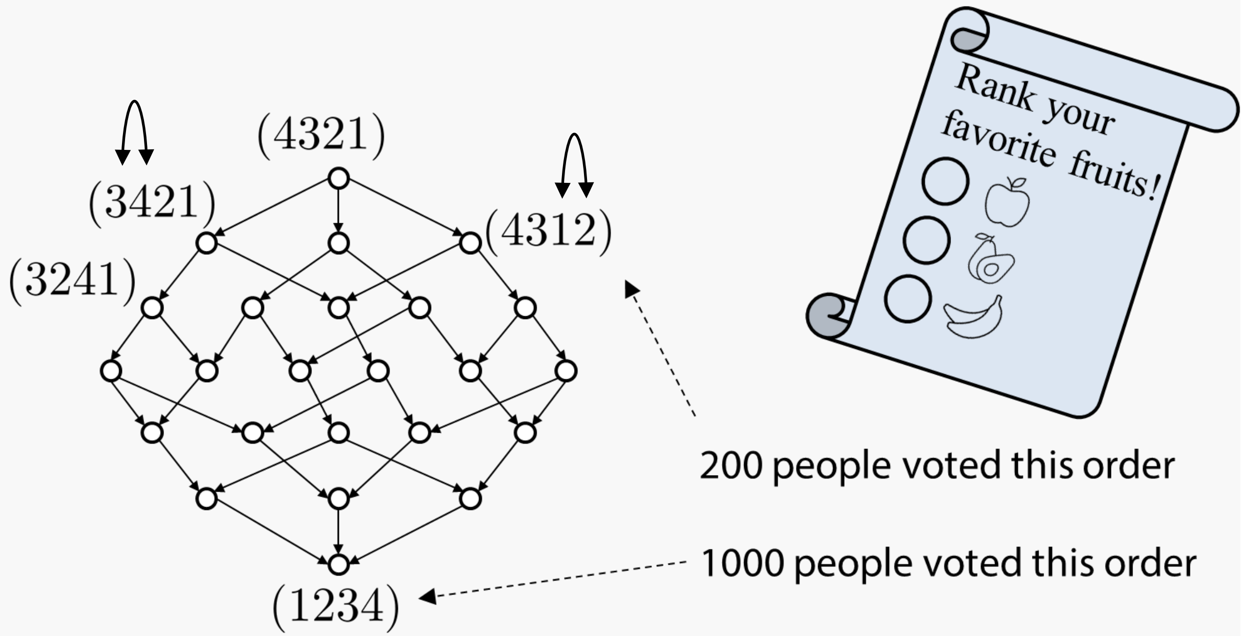
Permutation Lattice

Elements: *Permutations of length n*

b covers *a*: $a_i < a_{i+1}$

$$(b_1, \ldots, b_n) = (a_1, \ldots, a_{i+1}, a_i, \ldots, a_n)$$

Partial Order and Meet: *Derived from cover graph*



Filters: Linear Shift Invariant Systems

Shift(s) by $q \in L$

$$T_q \mathbf{s} = (s_{x \wedge q})_{x \in L}$$

Convolution

$$\mathbf{h} * \mathbf{s} = \left(\sum_{q \in L} h_q s_{x \wedge q} \right)_{x \in L}$$

Filter $\mathbf{h} = (h_q)_{q \in L}$

indexed by lattice

Shift Invariance ✓

Filters: Linear Shift Invariant Systems

Shift(s) by $q \in L$

$$T_q \mathbf{s} = (s_{x \wedge q})_{x \in L}$$

Convolution

$$\mathbf{h} * \mathbf{s} = \left(\sum_{q \in L} h_q s_{x \wedge q} \right)_{x \in L}$$

Filter $\mathbf{h} = (h_q)_{q \in L}$

indexed by lattice

Shift Invariance ✓

Fourier Transform diagonalizes all shifts and filters

$$\widehat{s}_y = \sum_{x \le y} \mu(x, y) s_x$$

indexed by lattice

$$\mu(x, x) = 1, \text{for } x \in L$$

$$\mu(x, y) = -\sum_{x \le z < y} \mu(x, z),$$

for $x \neq y$

Algebraic Lattice Theory

Filters: Linear Shift Invariant Systems

Shift(s) by $q \in L$

$$T_q \mathbf{s} = (s_{x \wedge q})_{x \in L}$$

Convolution

$$\mathbf{h} * \mathbf{s} = \left(\sum_{q \in L} h_q s_{x \wedge q} \right)_{x \in L}$$

Filter $\mathbf{h} = (h_q)_{q \in L}$

indexed by lattice

Shift Invariance ✓

Fourier Transform diagonalizes all shifts and filters

$$\widehat{s}_y = \textstyle\sum_{x \le y} \mu(x, y) s_x$$

indexed by lattice

$$\mu(x, x) = 1, \text{for } x \in L$$

$$\mu(x, y) = -\sum_{x \le z < y} \mu(x, z),$$

$$\text{for } x \neq y$$

As matrix (Discrete Lattice Transform)

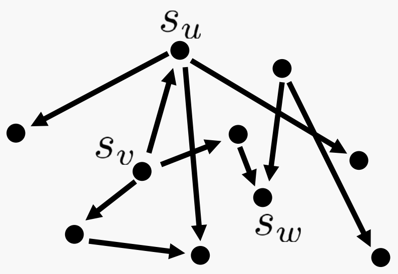
$$\text{DLT} = [\mu(x, y)]_{y, x \in L}$$

Algebraic Lattice Theory

Comparison Graph DSP

Graph DSP

Signals indexed by vertices of a graph



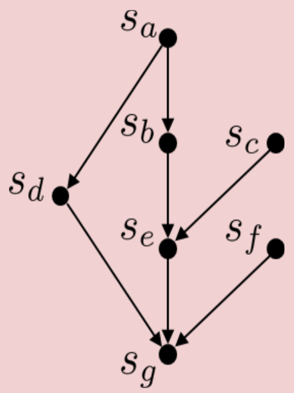
Shift captures adjacency structure

One generating shift
(adjacency or Laplacian)

Shift not always diagonalizable (digraphs)

New Discrete Lattice SP

Signals indexed by a meet/join lattice

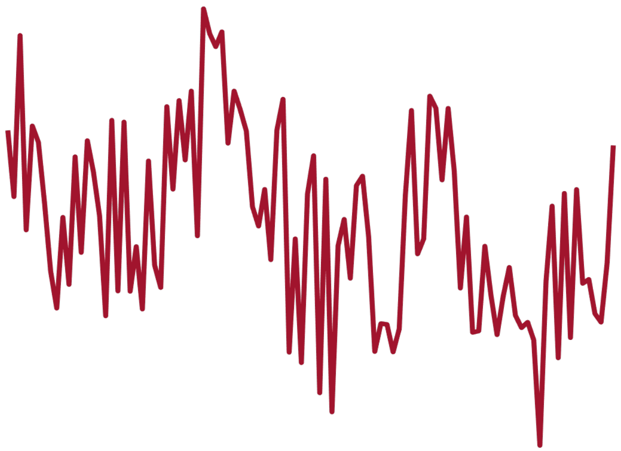


Shifts capture partial order structure

Several generating shifts
(one per 'maximal' element)

Shifts always diagonalizable

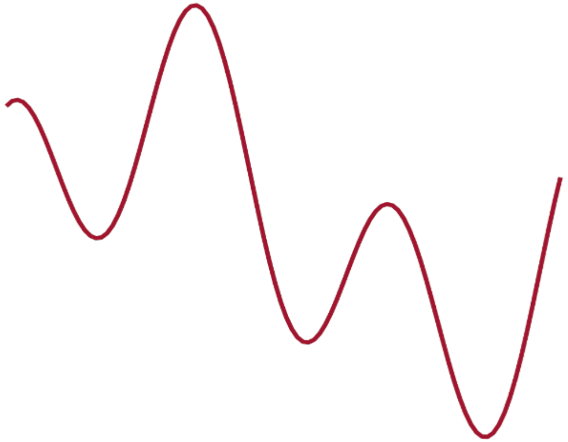
Wiener Filter

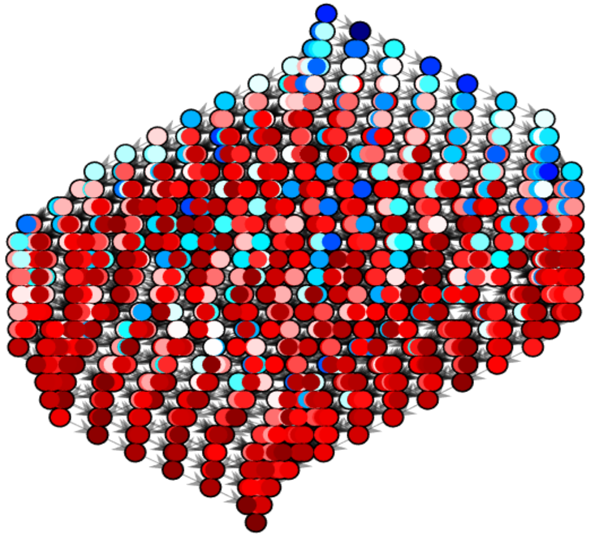


Classical SP

Wiener Filter

→

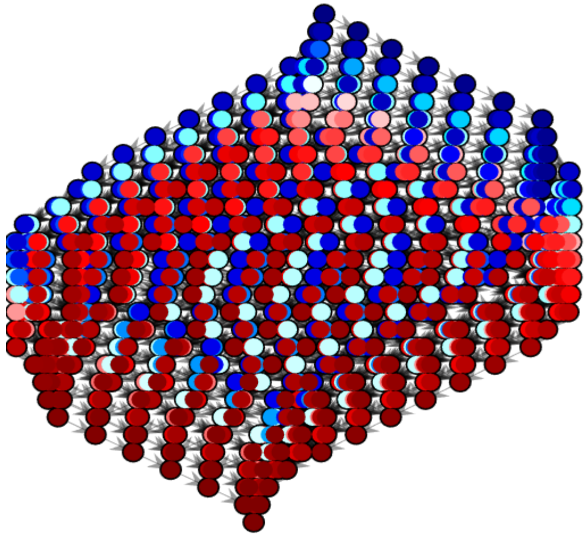




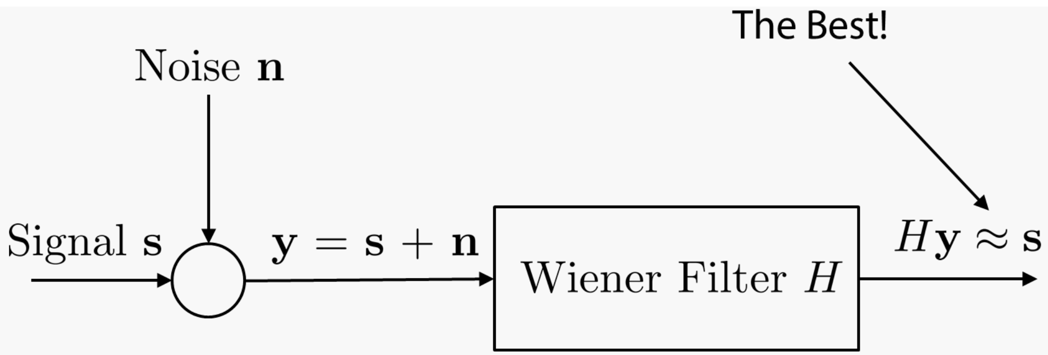
Our contribution

Wiener Filter for Lattice signals

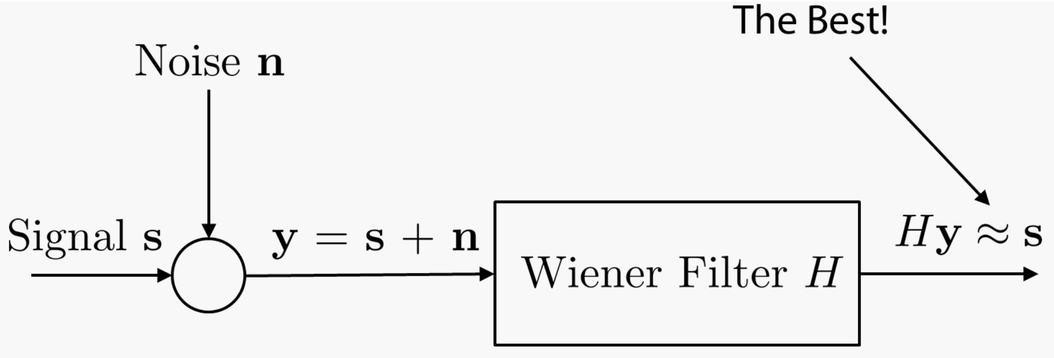
→



Wiener Filter



Wiener Filter



Classical SP

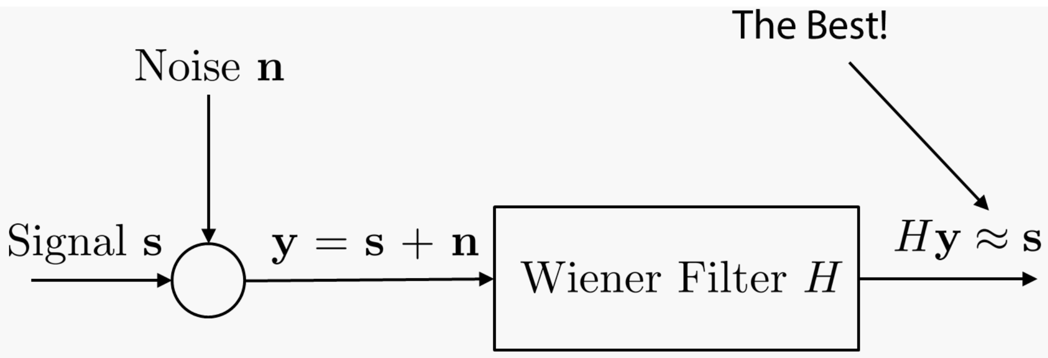
Shift

$$H \cdot y = \sum_{k=0}^{N} h_k T^k y$$

Filter Coefficients $\mathbf{h} = (h_0, \ldots, h_N)$

$$\min_{\mathbf{h}} \|H\mathbf{y} - \mathbf{s}\|_2^2$$

Wiener Filter



Classical SP

$$H \cdot y = \sum_{k=0}^{N} h_k T^k y$$

Filter Coefficients $\mathbf{h} = (h_0, \ldots, h_N)$

$$\min_{\mathbf{h}} \|H\mathbf{y} - \mathbf{s}\|_2^2$$

Lattice SP

One Shift? Yes!

$$H \cdot y = \sum_{k=0}^{N} h_k T_{\text{ep}}^k y$$

Filter Coefficients $\mathbf{h} = (h_0, \ldots, h_N)$

$$\min_{\mathbf{h}} \|H\mathbf{y} - \mathbf{s}\|_2^2$$

Energy-Preserving Shift

$$\mathrm{DLT} \cdot T_q \cdot \mathrm{DLT}^{-1} = \mathrm{diag}(i_y \mid i_y = 1 \text{ if } y \leq q, i_y = 0 \text{ else })$$

$$\mathrm{DLT} \cdot T_q \cdot \mathrm{DLT}^{-1} = p_q(\Lambda)$$

Diagonal, entries pairwise different

Energy-Preserving Shift

$$\mathrm{DLT} \cdot T_q \cdot \mathrm{DLT}^{-1} = \mathrm{diag}(i_y \mid i_y = 1 \text{ if } y \leq q, i_y = 0 \text{ else })$$

$$\mathrm{DLT} \cdot T_q \cdot \mathrm{DLT}^{-1} = p_q(\Lambda)$$

Diagonal, entries pairwise different

Choose

$$\Lambda_{\mathrm{ep}} = \mathrm{diag}(\exp(2\pi\,\mathrm{i}\,k/|L| \mid k = 0, \ldots, |L| - 1)$$

Energy-preserving shift

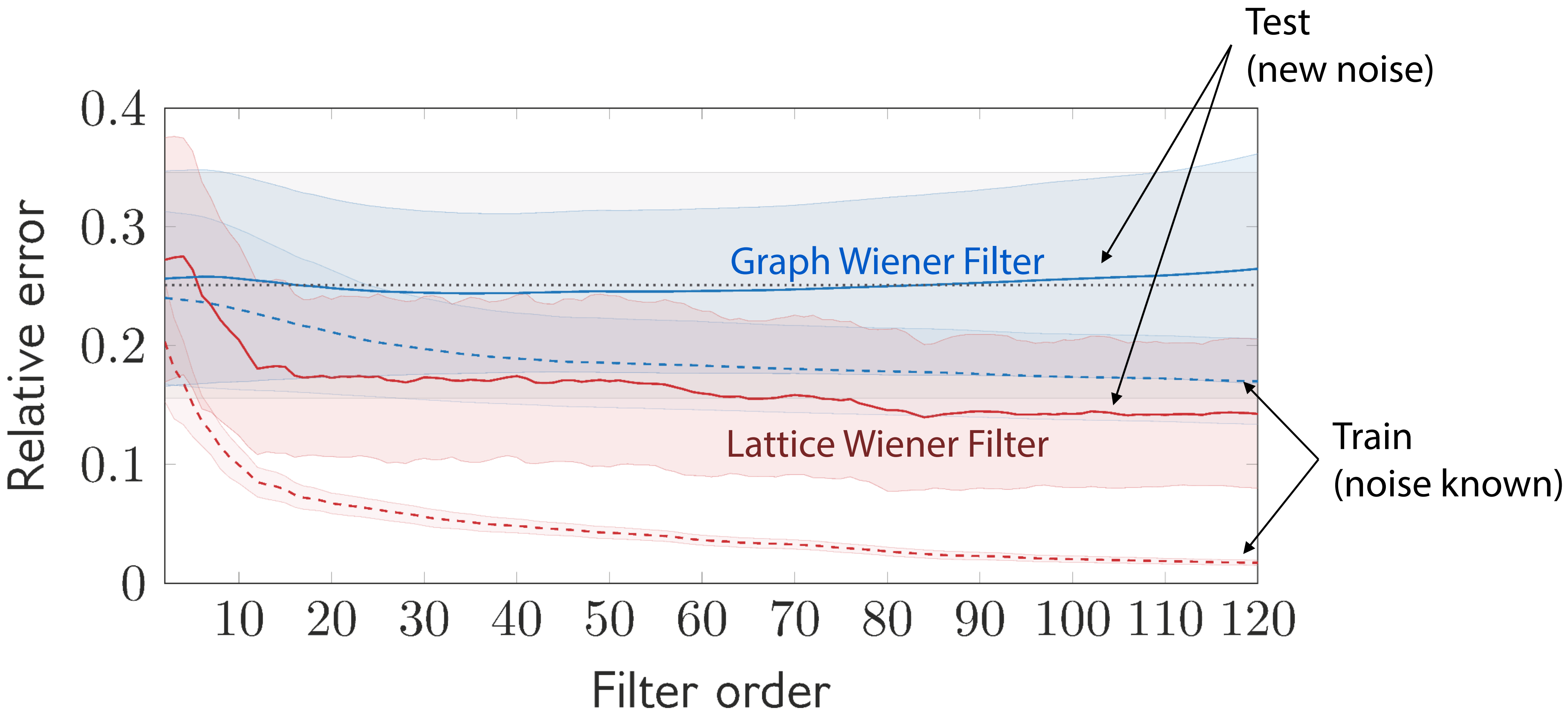
$$T_{\mathrm{ep}} = \mathrm{DLT}^{-1}\,\Lambda_{\mathrm{ep}}\,\mathrm{DLT}$$

Preserves energy

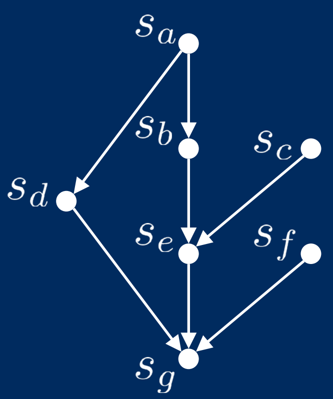
$$\|T_{\mathrm{ep}}\mathbf{s}\|_2 = \|\widehat{\mathbf{s}}\|_2$$

For Graphs: Gavili, Zhang 2017

Results (Spectrum Auctions)



Lattice Signal Processing



Convolution

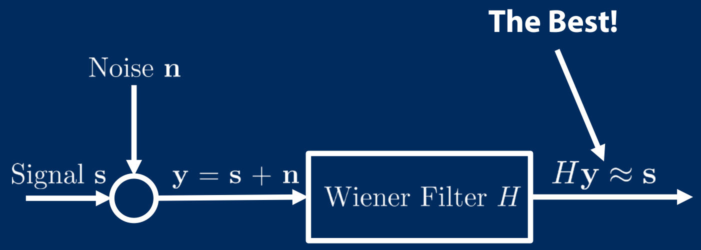
$$\mathbf{h} * \mathbf{s} = \left(\sum_{q \in L} h_q s_{x \wedge q}\right)_{x \in L}$$

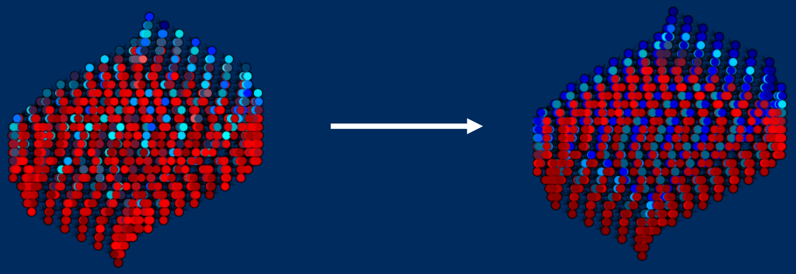
Fourier Transform

$$\widehat{s}_y = \sum_{x \leq y} \mu(x, y) s_x$$

https://acl.inf.ethz.ch/research/ASP/

Wiener Filter for Lattice Signals





Applications