WIENER FILTER ON MEET/JOIN LATTICES

Bastian Seifert, Chris Wendler, Markus Püschel

Department of Computer Science
ETH Zürich, Switzerland

ABSTRACT

Recent work introduced a framework for signal processing (SP) on meet/join lattices. Such a lattice is partially ordered and supports a meet (or join) operation that returns the greatest lower bound and the smallest upper bound of two elements, respectively. Lattices appear in various domains and can be used, for example, to express rankings in social choice theory or multisets in combinatorial auctions. Discrete lattice SP (DLSP) uses the meet operation as shift and derives associated notions of convolution and Fourier transform for signals indexed by lattices. In this paper we extend DLSP with Wiener filtering for denoising and demonstrate it on two prototypical applications.

***Index Terms*—** Meet/join lattice, partially ordered set, algebraic signal processing, combinatorial auction, ranked data

1. INTRODUCTION

Recent years have seen the advent of generalized signal processing (SP) frameworks. The most popular example is graph SP [1, 2], which generalizes classical SP concepts such as shift, shift-invariant filters and Fourier transform to signals indexed by the vertices of a graph. The availability of these concepts has brought new tools for processing or learning with data on sensor networks, biological networks, or point clouds [3, 4].

There are other non-Euclidean index domains besides graphs and first steps towards associated SP frameworks. Examples include simplicial complexes [5], powersets [6, 7], hypergraphs [8], and lattices [9–11]. A lattice is a partially ordered set with a meet and join operation that returns the greatest lower bound and smallest upper bound for any two elements, respectively. Data, or signals, on lattices naturally occur in various areas including ranked data in social choice theory [12], bidders' value functions in combinatorial auctions [13], statistical data on formal concepts [14] or genotype-phenotype mappings in computational biology [15].

Lattice SP [9, 11] uses the meet (or join) operation to define a shift operation and then uses the general procedure from algebraic signal processing [16] to obtain an associated notion of convolution, Fourier transform, and frequency response.

Contribution. In this paper, we extend lattice SP with a suitable notion of Wiener filtering. Inspired by Wiener filtering on graphs [17], we first define an energy-preserving shift on lattices, which is then used for the construction of Wiener filters, which are polynomials in this shift. We show two prototypical application examples for denoising lattice signals in the context of combinatorial auctions and ranked data.

2. DISCRETE LATTICE SIGNAL PROCESSING

We provide the necessary background on lattice theory [18] and discrete lattice SP (DLSP) following [9].

Poset. A finite set $\mathcal{L}$ is a *partially ordered set* (poset) if it is equipped with a binary relation $\leq$ that satisfies for the elements $a, b, c \in \mathcal{L}$

1. reflexivity, $a \leq a$,
2. antisymmetry, $a \leq b$ and $b \leq a$ implies $a = b$, and
3. transitivity, $a \leq b$ and $b \leq c$ implies $a \leq c$.

Note that not all elements in $\mathcal{L}$ need to be comparable. For example, the powerset (set of all subsets) of a finite set is a poset with $\leq = \subseteq$.

Semilattice. A (meet-)semilattice is a poset with a meet operation that returns the unique greatest lower bound (or *meet*) $a \wedge b$ of two elements $a, b \in \mathcal{L}$. This means for all c with $c \leq a$ and $c \leq b$ also $c \leq a \wedge b$ holds.

For example, the powerset of a finite set is a semilattice with $\wedge = \cap$ (intersection).

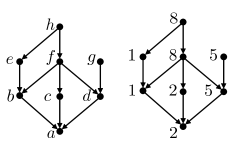
An analogous definition considers the join $\vee$, which returns the smallest upper bound but is not needed in this paper. Here, we will say lattice to mean meet-semilattice.

Cover graph. We write $a < b$ if $a \leq b$ and $a \neq b$. An element b *covers* a, if $a < b$ and no element is in between. A lattice can be visualized by its cover graph with nodes $V = \mathcal{L}$ and edges $E = \{(b, a) \mid b \text{ covers } a\}$. It is usually drawn from top (greater elements) to bottom (smaller elements).

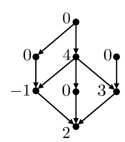
The cover graph of an example lattice $\mathcal{L} = \{a, \dots, h\}$ with 8 elements is shown in Fig. 1a. It shows that, e.g., $g \wedge f = d$, $e \wedge d = a$, $e \wedge b = b$. The element a is the unique minimum, the meet of all elements.

Lattice signal. Given a lattice $\mathcal{L}$ with n elements, a lattice signal $\mathbf{s}$ associates a value with each of its elements. Formally,

$$\mathbf{s} = (s_a)_{a \in \mathcal{L}} \in \mathbb{C}^n.$$



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & -1 & -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 1 \end{bmatrix}$$



(a) $\mathcal{L}$ (b) $\mathbf{s}$ (c) DLT (d) $\widehat{\mathbf{s}}$

Fig. 1: Meet-semilattice $\mathcal{L}$, example lattice signal $\mathbf{s}$, lattice Fourier transform DLT, and $\widehat{\mathbf{s}}$. The rows and columns of F are indexed by the lattice elements in alphabetical order.

To make it a vector, we choose a so-called *topological ordering* from small to large elements. An example signal is shown in Fig. 1b.

Lattice shifts. For each $x \in \mathcal{L}$, DLSP defines an associated linear shift from the meet operation, which can be expressed as a matrix T_x:

$$T_x\mathbf{s} = (s_{a\wedge x})_{a\in\mathcal{L}} \tag{1}$$

Note that the shifts are idempotent, i.e., $T_x^2 = T_x$.

Filters and convolution. As usual, filters are linear combinations of shifts, which yields the associated notion of convolution. Namely, if $\mathbf{h} \in \mathbb{C}^n$, then

$$\mathbf{h} * \mathbf{s} = \Big(\sum_{x\in\mathcal{L}} h_x T_x\Big)\mathbf{s} = \Big(\sum_{x\in\mathcal{L}} h_x s_{a\wedge x}\Big)_{a\in\mathcal{L}}. \tag{2}$$

Filters are shift-invariant, $\mathbf{h} * T_x\mathbf{s} = T_x(\mathbf{h}*\mathbf{s})$, since the meet-operation is commutative.

Fourier transform. The Fourier transform, called discrete lattice transform (DLT) simultaneously diagonalizes all filters [11], i.e., all shift matrices T_x. It is obtained from lattice theory [19], and defined as

$$\widehat{\mathbf{s}} = \mathrm{DLT}_{\mathcal{L}}\,\mathbf{s}, \quad \mathrm{DLT}_{\mathcal{L}} = [\mu(x,y)]_{y,x\in\mathcal{L}}.$$

Here, μ is the Moebius function, which can be computed recursively via

$$\mu(x,x) = 1 \quad \text{and} \quad \mu(x,y) = -\sum_{x\le z<y} \mu(x,z). \tag{3}$$

The DLT for the lattice in Fig. 1a is shown in Fig. 1c, and the spectrum $\widehat{\mathbf{s}}$ of $\mathbf{s}$ in Fig. 1b in Fig. 1d. Note that the DLT is not orthogonal and always of triangular shape (with topological ordering of $\mathcal{L}$).

The inverse Fourier transform is given by $\mathrm{DLT}_{\mathcal{L}}^{-1} = [\iota_{y\le x}]_{x,y\in\mathcal{L}}$, where $\iota_{y\le x} = 1$ if $y \le x$ and $\iota_{y\le x} = 0$ otherwise.

Fast algorithms for computing the DLT exist [20].

Difference to graph SP. The meet shifts in DLSP operate differently from the adjacency shift in [21], capturing the partial order structure rather than adjacency in the cover graph. Also note that cover graphs are directed and acyclic and thus the adjacency matrix has 0 as the only eigenvalue, a problem in graph SP [3, Sec.III-A].

3. ENERGY-PRESERVING SHIFT AND WIENER FILTERS

In this section we port Wiener filtering to DLSP. The basic idea is to first define a shift operator that preserves energy in the frequency domain and then design Wiener filters as polynomials in this shift. This high level approach is analogous to the one in [17] for graph SP, but instantiated for lattices. The idea of energy-preserving shift addresses the problem that the natural shift operators (e.g., adjacency shift in graph SP or lattice shifts here) do not have this property, unlike the (translation) shift underlying classical discrete-time SP.

Energy-preserving shift. We define the energy-preserving shift as

$$T_{\text{ep}} = \mathrm{DLT}^{-1} \cdot \Lambda_{\text{ep}} \cdot \mathrm{DLT}, \tag{4}$$

where $\Lambda_{\text{ep}} = \mathrm{diag}_{0\le k<n}(\exp(-2\pi jk/n)$, i.e., we force the same frequency response as the standard cyclic shift has. By construction, T_{ep} preserves energy in the frequency domain: $\|\widehat{T_{\text{ep}}\mathbf{s}}\|_2 = \|\Lambda_{\text{ep}}\widehat{\mathbf{s}}\|_2 = \|\widehat{\mathbf{s}}\|_2$.

Lattice Wiener filters. Wiener filtering designs an optimal denoising filter from a noisy version of a known reference signal. This filter can then be used to denoise similar signals that are noisy w.r.t. the same noise model.

Consider the reference signal $\mathbf{s}$ and a noisy measurement $\mathbf{y} = \mathbf{s} + \mathbf{n}$. The Wiener filter of order N based on the energy-preserving shift, mimicking [17], has the (matrix) form

$$H = \sum_{k=0}^{N} h_k T_{\text{ep}}^k, \tag{5}$$

where the filter coefficients $\mathbf{h} = (h_0, \dots, h_N)$ are the solution of the minimization problem

$$\min_{\mathbf{h}} \|H\mathbf{y} - \mathbf{s}\|_2^2. \tag{6}$$

Thus, the Wiener filter H is the optimal denoising filter with respect to the Euclidean error. Note that the powers of T_{ep} can be computed efficiently in the frequency domain: $T_{\text{ep}}^k = \mathrm{DLT}^{-1}\,\Lambda_{\text{ep}}^k\,\mathrm{DLT}$.

Equation (6) is equivalent to

$$\min_{\mathbf{h}} \|B\mathbf{h} - \mathbf{s}\|_2^2, \quad \text{with } B = [\mathbf{y}\ T_{\text{ep}}\mathbf{y}\ \dots\ T_{\text{ep}}^N\mathbf{y}], \tag{7}$$

and can be solved by setting the gradient of $\|B\mathbf{h}-\mathbf{s}\|_2^2$ to zero and solving for $\mathbf{h}$, which is equivalent to solving the linear system

$$B^H B\mathbf{h} = B^H\mathbf{s}. \tag{8}$$

4. EXPERIMENTS

In this section, we first introduce a noise model suitable for lattice signals and then evaluate our Wiener filters on two prototypical types of lattice signals.

Lattice white noise. White noise has equal intensity across frequencies. In discrete time SP, white noise can be simulated by sampling a Gaussian noise vector with independent components and adding it to the signal. Doing so with a lattice signal is not equivalent to white noise since the DLT is not orthogonal. Thus, we simulate white noise directly in the frequency domain.

The same phenomenon occurs in graph SP with an irregularity-aware Fourier transform [22].

Experimental setup. To properly evaluate our Wiener filters and exclude overfitting effects, we compute the filter on a *known* reference signal $\mathbf{s}^{\text{ref}}$ and its noisy version $\mathbf{y}^{\text{ref}} = \mathbf{s}^{\text{ref}} + \mathbf{n}^{\text{ref}}$, where $\mathbf{n}^{\text{ref}} = \text{DLT}_{\mathcal{L}}^{-1}\, \widehat{\mathbf{n}}^{\text{ref}}$ is our lattice white noise, i.e., $\widehat{\mathbf{n}}^{\text{ref}}$ is sampled from a zero-mean isotropic Gaussian distribution. Then, we evaluate the learnt filter on *unknown* noisy test signals $\mathbf{y}^{\text{test}} = \mathbf{s}^{\text{test}} + \mathbf{n}^{\text{test}}$ with a different noise sample from the same noise model.

Benchmark. As benchmark for the lattice Wiener filter, we use a graph Wiener filter on the cover graph. Doing so with the directed adjacency matrix (shift) A produced meaningless results since B in (8) becomes rank-deficient. Hence we use also here the energy-preserving shift [17]. Since it requires A to be diagonalizable, and A has only eigenvalue 0 with typically Jordan blocks of size greater than 1, we use the undirected cover graph instead. Note that the comparison puts the benchmark at a disadvantage, since lattice white noise cannot be modulated with graph filters.

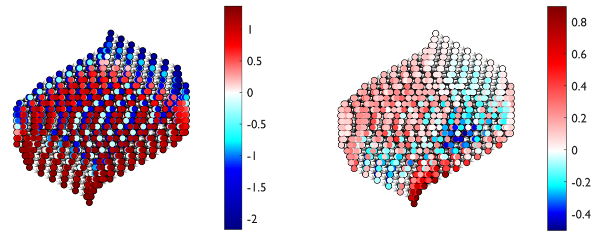
4.1. Spectrum auctions

Spectrum auctions [23] are combinatorial auctions [13] in which licenses for bands of the electromagnetic spectrum are sold to a set of bidders (e.g., telecommunication companies). For each frequency band, multiple licenses may exist, making the set of available licenses a multiset (a set that allows for duplicate elements). Bidders submit bids for submultisets of licenses. Thus, each bidder can be modeled as a multiset function (i.e., a mapping from multisets to positive values) called value function. As explained next, the set of all submultisets is a lattice, making value functions lattice signals.

Lattice. Formally, we represent the multiset of all available licenses with a vector $m \in \mathbb{N}_0^f$, where the entry $m_i > 0$ is the number of available licenses of the ith frequency band, $1 \leq i \leq f$. We call $a \in \mathbb{N}_0^f$ a submultiset of m if $a \leq m$, defined as $a_i \leq m_i$ for all $i \in \{1, \dots, f\}$. The set of all submultisets $\mathcal{L} = \{a \in \mathbb{N}_0^f : a \leq m\}$ is a lattice with meet $a \wedge b = \min(a, b)$, where the minimum is taken componentwise. Thus, $|\mathcal{L}| = \prod_{1 \leq i \leq f}(m_i + 1)$.

Signal. Given a bidder, its value function assigns values $s_a \in \mathbb{R}^+$ to all submultisets of licenses $a \in \mathcal{L}$ and, thus, is a lattice signal $\mathbf{s} = (s_a)_{a \in \mathcal{L}}$.

Experiment. As common in this field (e.g., [24]), we rely on simulated bidders. To obtain a reference and a test



(a) (b)

Fig. 2: (a) The normalized bidder signal $\mathbf{s}^{\text{ref}}$ on which the filter coefficients are calculated, and (b) a sample $\mathbf{n}^{\text{ref}}$ of lattice white noise, which is used to corrupt the signal.

signal, we simulate bidder valuation functions using the so-called single region value model (SRVM) from the spectrum auctions test suite [25]. In SRVM there three bands with multiplicities 6, 14 and 9, i.e., $m = (6, 14, 9)^T$ and $|\mathcal{L}| = 7 \cdot 15 \cdot 10 = 1050$, and four different parameterized bidder types (small, high, primary, and secondary). Bidders are sampled by randomly sampling their parameters. For our reference signal we sampled one secondary bidder $\mathbf{s}^{\text{ref}}$ and for our test signal one primary bidder $\mathbf{s}^{\text{test}}$. We show $\mathbf{s}^{\text{ref}}$ and a sample $\mathbf{n}^{\text{ref}}$ of lattice white noise in Fig. 2. Note that the (non-orthogonal) inverse Fourier transform $\text{DLT}_{\mathcal{L}}^{-1}$ sums up the smaller elements of each entry, which explains the visible structure. For example, the value of $\mathbf{n}^{\text{ref}}$ indexed by the maximal element has a value near zero since it is the sum of all elements in a zero-mean isotropic Gaussian vector.

Result. In Fig. 3, we evaluate our Wiener filters in terms of relative reconstruction error on the reference (dashed) and test bidder (solid) for increasing filter order N. We compare lattice Wiener filters (red) with graph Wiener filters (blue). We add lattice white noise as defined above to both signals such that the signal-to-noise ratio is 12.5 ± 2.7 dB, compute the Filter coefficients on the reference signal (Fig. 2a) and its corrupted version, and apply the resulting filter to the corrupted test signal. To obtain mean (lines) and standard deviations (shaded areas), we repeated the experiment 100 times.

We observe that the lattice Wiener filters denoise the test signal and that the standard deviation stays within the range of the standard deviation from the noise. Beyond $N = 40$, the error curve flattens. For the reference signal, the error continues to decrease and the standard deviation goes to zero, a sign of (expected) overfitting. The graph Wiener filters on the other hand perform poorly, probably because they cannot properly modulate the lattice white noise.

4.2. Rankings

Ranking data is used in social choice theory [26] to model preferences of human groups. In its simplest form ranked data is obtained when individuals have to rank a set of choices. Each possible ranking corresponds to a permutation of the set

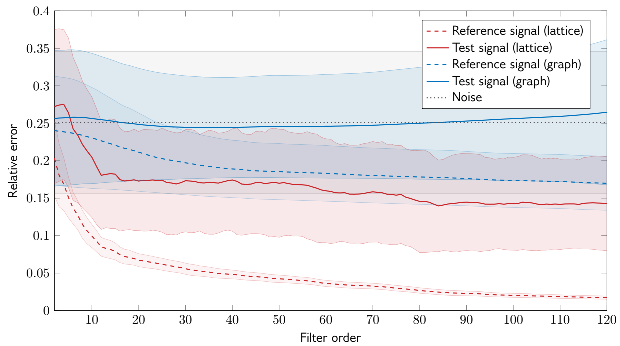


Fig. 3: Denoising a bidder signal with lattice (red) and graph Wiener filter (blue) of different order. The results on the reference signals in both cases are shown dashed. The shaded areas are the standard deviations over 100 simulations.

of choices and can be associated with the number of individuals that ranked the choices accordingly. As we explain next, the set of all permutations is a lattice, which makes the individual counts a lattice signal.

Lattice. Formally, the permutations of a list $(1, \ldots, n)$ with a suitable initial ordering of n choices form a lattice $\mathcal{L}_n$ [27] of size $n!$. A permutation $b = (b_1, \ldots, b_n)$ covers the permutation $a = (a_1, \ldots, a_n)$ iff there exists a transposition τ_i that exchanges two consecutive elements in positions $i, i+1$, such that

$$b = a \cdot \tau_i = (a_1, \ldots, a_{i-1}, a_{i+1}, a_i, a_{i+2}, \ldots, a_n) \quad (9)$$

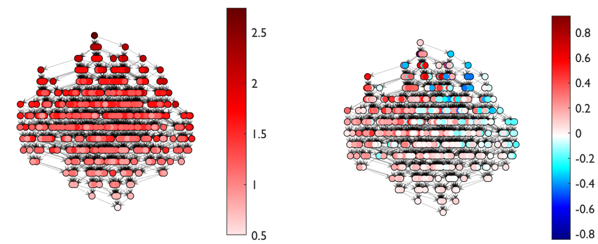
The partial order $\leq$ and the meet $\wedge$ are now derived from the cover graph. The original $(1, \ldots, n)$ is the minimum and we have $a \leq b$ if there is a path from b to a, i.e., if there exists a sequence of transpositions such that $b = a \cdot \tau_{i_1} \cdots \tau_{i_k}$. The meet $a \wedge b$ is the first element in $\mathcal{L}$ that can be reached from both a and b (using transpositions). E.g., $(1, 2, 3) \leq (1, 3, 2)$ and $(2, 3, 1) \wedge (1, 3, 2) = (1, 2, 3)$ as $(2, 3, 1) = (1, 2, 3) \cdot \tau_1 \cdot \tau_2$ and $(1, 3, 2) = (1, 2, 3) \cdot \tau_2$. The maximal element in $\mathcal{L}_n$ is $(n, \ldots, 1)$.

Signal. We consider synthetic low-frequency signals with decaying spectrum as explained next.

Experiment. We consider the ranking lattice $\mathcal{L}_6$ containing $6! = 720$ permutations/rankings and create synthetic low frequency signals. Specifically, we set for every $b \in \mathcal{L}$, $\widehat{s}_b = \alpha^\ell$, where $\alpha < 1$ and ℓ is the distance of b from the minimum in the cover graph. For the reference signal $\mathbf{s}^{\text{ref}}$ we set $\alpha = 0.5$. We create two test signals $\mathbf{s}^{\text{test}_1}$ and $\mathbf{s}^{\text{test}_2}$ with $\alpha = 0.45$ and $\alpha = 0.55$, respectively.

Fig. 4 shows the reference signal $\mathbf{s}^{\text{ref}}$ and a sample $\mathbf{n}^{\text{ref}}$ from our noise distribution.

Results. In Fig. 5, we evaluate our Wiener filters in terms of relative reconstruction error on the reference (dashed) and test signals (solid) for increasing filter order N. Again we compare lattice Wiener filters (red) with graph Wiener filters (blue). We add lattice white noise as defined above to both



(a) (b)

Fig. 4: (a) The ranking signal $\mathbf{s}^{\text{ref}}$ on which the Wiener filter coefficients are calculated, and (b) a sample $\mathbf{n}^{\text{ref}}$ of lattice white noise used to corrupt the signal.

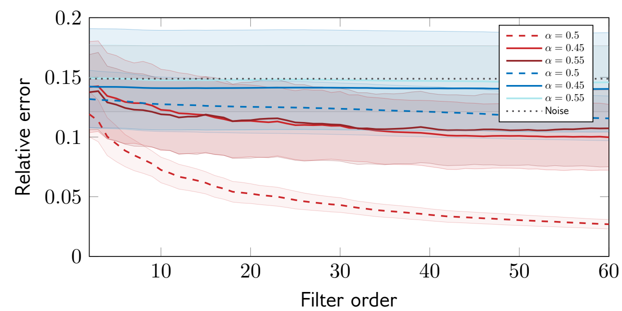


Fig. 5: Denoising synthetic signals on the permutation lattice with lattice (red) and graph Wiener filter (blue). The results on the reference signals in both cases are shown dashed. The shaded areas are the standard deviations over 100 simulations.

signals such that the signal-to-noise ratio is 17 ± 1.3 dB, compute the Filter coefficients on the reference signal (Fig. 4a) and its corrupted version, and apply the resulting filter to the corrupted test signals. In order to obtain means (lines) and standard deviations (shaded areas), we repeated the experiment 100 times. The results are qualitatively similar as in the previous experiment.

5. CONCLUSION

We have expanded discrete-lattice SP with a suitable definition and construction of Wiener filters. We thus add to the body of work that shows how SP methods can be ported and used on signals with different index domains. In particular, and similar to the situation in SP on directed graphs, this is possible even if the Fourier transform is not orthogonal.

As a proof of concept, we performed two prototypical denoising experiments, also showing that graph Wiener filters cannot help with lattice white noise. The construction and application of Wiener filters is computationally efficient and should easily scale to lattices with a few thousands elements. Future work should go deeper into application domains to put the filters to real-world use.

6. REFERENCES

[1] D. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[2] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[3] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[4] M. M. Bronstein, J. Bruna, Y. LeCun, Szlam A., and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, pp. 18–42, 2017.

[5] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Trans. Signal Process.*, vol. 68, pp. 2992–3007, 2020.

[6] Markus Püschel, "A discrete signal processing framework for set functions," in *Proc. Int. Conf. Acoust., Speech, and Signal Process. (ICASSP)*, 2018, pp. 1935–1968.

[7] Markus Püschel and Chris Wendler, "Discrete signal processing with set functions," *IEEE Trans. Signal Proc.*, 2021, to appear.

[8] S. Zhang, Z. Ding, and S. Cui, "Introducing hypergraph signal processing: Theoretical foundation and practical applications," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 639–660, 2020.

[9] M. Püschel, "A discrete signal processing framework for meet/join lattices with applications to hypergraphs and trees," in *Proc. Int. Conf. Acoust., Speech, and Signal Process. (ICASSP)*, 2019, pp. 5371–5375.

[10] C. Wendler and M. Püschel, "Sampling signals on meet/join lattices," in *Proc. Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2019, pp. 1–5.

[11] M. Püschel, B. Seifert, and C. Wendler, "Discrete signal processing on meet/join lattices," arXiv preprint arXiv:2012.04358, 2020.

[12] B. Monjardet and V. Raderanirina, "Lattices of choice functions and consensus problems," *Social Choice and Welfare*, vol. 23, no. 3, pp. 349–382, 2004.

[13] L. M. Ausubel, P. Cramton, and P. Milgrom, "The clock-proxy auction: A practical combinatorial auction design," in *Combinatorial Auctions*, pp. 115–138. MIT Press, 2006.

[14] B. Ganter and R. Wille, *Formal concept analysis: mathematical foundations*, Springer Science & Business Media, 2012.

[15] N. Beerenwinkel, P. Knupfer, and A. Tresch, "Learning monotonic genotype-phenotype maps," *Stat Appl Genet Mol Biol*, vol. 10, no. 1, pp. Article 3, 2011.

[16] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3572–3585, 2008.

[17] A. Gavili and X.-P. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6303–6318, 2017.

[18] G. Grätzer, *Lattice Theory: Foundation*, Birkhäuser, 2011.

[19] G.-C. Rota, "On the foundations of combinatorial theory. I. theory of Möbius functions," *Z. Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 2, no. 4, pp. 340–368, 1964.

[20] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, J. Nederlof, and P. Parviainen, "Fast zeta transforms for lattices with few irreducibles," *ACM Trans. Algorithms*, vol. 12, no. 1, pp. 4:1–4:19, 2015.

[21] A. Sandryhaila, J. Kovacevic, and M. Püschel, "Algebraic signal processing theory: 1-D nearest-neighbor models," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2247–2259, 2012.

[22] B. Girault, A. Ortega, and S. S. Narayanan, "Irregularity-aware graph Fourier transforms," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5746–5761, 2018.

[23] P. Cramton, "Spectrum auction design," *Review of Industrial Organization*, vol. 42, no. 2, pp. 161–190, 2013.

[24] J. Weissteiner and S. Seuken, "Deep learning-powered iterative combinatorial auctions," in *Proc. AAAI Conf.of Artificial Intelligence*, 2020, pp. 2284–2293.

[25] M. Weiss, B. Lubin, and S. Seuken, "SATS: A universal spectrum auction test suite," in *Proc. Conf. Autonomous Agents and MultiAgent Systems (AAMAS)*, 2017, pp. 51–59.

[26] N. Mattei and T. Walsh, "PrefLib: A library of preference data," in *Proc. Int. Conf. Algorithmic Decision Theory (ADT)*, 2013, pp. 259–270.

[27] G. Markowsky, "Permutation lattices revised," *Mathematical Social Sciences*, vol. 27, no. 1, pp. 59–72, 1994.