Maximizing supermodular functions on product lattices, with application to maximum constraint satisfaction*

Andrei Krokhin
Department of Computer Science
University of Durham,
Durham, DH1 3LE, UK
andrei.krokhin@durham.ac.uk

Benoit Larose
Department of Mathematics and Statistics
Concordia University
Montréal, Canada H3G 1M8
larose@mathstat.concordia.ca

Abstract

Recently, a strong link has been discovered between supermodularity on lattices and tractability of optimization problems known as maximum constraint satisfaction problems. The present paper strengthens this link. We study the problem of maximizing a supermodular function which is defined on a product of n copies of a fixed finite lattice and given by an oracle. We exhibit a large class of finite lattices for which this problem can be solved in oracle-polynomial time in n. We also obtain new large classes of tractable maximum constraint satisfaction problems.

Keywords: supermodular function, lattices, optimization, tractability, constraint satisfaction.

AMS subject classification: 90C27, 68Q25, 68T20.

1 Introduction

Sub- and supermodular set functions are special real-valued functions defined on the powerset of a set. They are well studied in combinatorics (see bibliographical survey [10]), and have numerous applications in combinatorial optimization [9, 11] and elsewhere (see, e.g., [24, 28]). Minimization of submodular set functions is one of most well known tractable problems in combinatorial optimization [11, 27]. Examples of other combinatorial problems that can be solved by using submodular set function minimization include the minimum s-t cut problem in networks and finding the largest common independent set in two matroids. The submodular set function minimization problem was also considered for various relaxations of submodular functions such as intersecting or crossing submodular functions. Such relaxations are either defined on some family of subsets of a set, or use a slightly modified form of submodularity [27].

A more general form of sub- and supermodularity is the one where functions are defined on general (i.e., algebraic) lattices [28]. This form of supermodularity is very popular in financial and actuarial mathematics (see, e.g., supermodular games [28] and supermodular order on multivariate distributions [23, 25]), but it is relatively little studied in combinatorics and optimization (see, e.g., Section 60.3a of [27]), except for the special case when the lattice is a chain, i.e., a totally ordered set. Sub- and supermodular functions on finite chains can be alternatively represented by matrices and arrays which are called Monge and anti-Monge, respectively, matrices and arrays. Such matrices and arrays are used to identify tractable cases of hard optimization problems such as TSP [1].

*A preliminary version of some parts of this paper appears in Proceedings of CP'05, Sitges, Spain, 2005.

Recently, the general form of supermodularity (on lattices) has been applied to classify the complexity of maximum constraint satisfaction problems which are very actively studied optimisation problems in artificial intelligence [6] and complexity theory [5]. In a maximum constraint satisfaction problem, informally speaking, one is given a finite collection of constraints on overlapping sets of variables, and the goal is to find an assignment of values to the variables with a maximum number (or total weight) of satisfied constraints. Some recent papers [3, 8, 19] discovered that there is a strong link between supermodularity on lattices and tractability of maximum constraint satisfaction problems with a restricted set of allowed constraints, and developing this connection is the main aim of the present paper.

2 Sub- and supermodularity, and lattices

In this section we describe the problem of maximizing a supermodular function on a product lattice.

Definition 2.1 *Let A be finite set. A function $f : 2^A \to \mathbb{R}$ is called a* supermodular set function *(on A) if the inequality*

$$f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y)$$

holds for all $X, Y \subseteq A$, and it is called submodular *if the inverse inequality holds for all X, Y.*

The *submodular function minimization problem* is, given a submodular function f on A, to find a subset $X \subseteq A$ with minimum $f(X)$. It is known [11, 17, 16, 26, 27] that a submodular function on a set A can be minimized in polynomial-time (in $|A|$) provided getting a value of f is a primitive operation.

A partial order on a set D is called a *lattice order* if, for every $x, y \in D$, there exists a greatest lower bound $x \sqcap y$ and a least upper bound $x \sqcup y$. The corresponding algebra $\mathcal{L} = (D, \sqcap, \sqcup)$ is called a *lattice*. A subset of D closed under the operations $\sqcap$ and $\sqcup$ is called a *sublattice* of $\mathcal{L}$. If $\mathcal{L}_i$ is a lattice on D_i, $i = 1, \ldots, n$, then the *product lattice* $\mathcal{L}_1 \times \ldots \times \mathcal{L}_n$ is a lattice with base set $D_1 \times \ldots \times D_n$ and operations acting component-wise. The lattice $\mathcal{L}^n$ is the (direct) product of n copies of $\mathcal{L}$, and it is known as the n-th *power* of $\mathcal{L}$. For more information about lattices, see [13].

Definition 2.2 *Let $\mathcal{L}$ be a lattice on D. A function $f : D^n \to \mathbb{R}$ is called* supermodular *on $\mathcal{L}$ if*

$$f(\mathbf{a}) + f(\mathbf{b}) \leq f(\mathbf{a} \sqcap \mathbf{b}) + f(\mathbf{a} \sqcup \mathbf{b}) \text{ for all } \mathbf{a}, \mathbf{b} \in \mathcal{L}^n.$$

and f is called submodular *on $\mathcal{L}$ if the inverse inequality holds.*

If A is a set with $|A| = n$ then, by identifying subsets of A with 0-1 n-tuples, one can easily check that the submodular (set) functions on A are simply the n-ary submodular functions on a lattice on $\{0, 1\}$ with order $0 < 1$. Therefore, it is natural to consider the problem of minimizing the submodular functions on a given fixed finite lattice $\mathcal{L}$, in the following form:

Instance: A number $n \geq 1$ and a submodular function f on $\mathcal{L}^n$.

Goal: Find an element $\mathbf{a} \in \mathcal{L}^n$ such that $f(\mathbf{a}) = \min\{f(\mathbf{b}) \mid \mathbf{b} \in \mathcal{L}^n\}$.

We will denote this problem by SFM($\mathcal{L}$). We will say that SFM($\mathcal{L}$) is *oracle-tractable* if it can be solved in polynomial time in n (provided getting the value of f on a tuple is a primitive operation). It was mentioned in [3] as an open question whether SFM($\mathcal{L}$) is oracle-tractable for any fixed lattice $\mathcal{L}$. This question was motivated in [3] by its applications in constraint satisfaction which we will describe in the next section.

Note that a function f is submodular if and only if $-f$ is supermodular. Therefore, the SFM($\mathcal{L}$) problem can also be understood as the *supermodular function maximization problem.* We will always use this reformulation because we will later apply algorithms for SFM($\mathcal{L}$) to solve certain maximization problems.

Recall that a lattice is called *distributive* if it can be represented by subsets of a set A, where the operations $\sqcap$ and $\sqcup$ are interpreted as set-theoretic intersection and union, respectively. Note that, in some earlier papers on submodular functions, a family of subsets closed under intersection and union is called simply a lattice family (or a ring family). The following result is proved in Section 49.3 of [27] (see also [26]).

Theorem 2.3 *A submodular function defined on a lattice family L on a set A can be minimized in polynomial time in $|A|$ provided we can compute in polynomial time the largest and the smallest sets in the family, and the pre-order $\preceq$ on A defined as follows: $u \preceq v$ if and only if each set $U \in L$ containing v also contains u.*

Assume that we *fix* a finite distributive lattice $\mathcal{L}$. It is well-known that $\mathcal{L}$ can be represented by subsets of a set A such that $|A| \leq |\mathcal{L}|$. Clearly, we can compute in constant time the sets (as 0-1 $|A|$-tuples) representing the largest and the smallest elements of $\mathcal{L}$, and we can also compute the pre-order $\preceq$ in constant time. Obviously, for any n, the lattice $\mathcal{L}^n$ can be represented by subsets of a set B of cardinality $n|A|$ (since an element of $\mathcal{L}$ is represented by using $|A|$ bits), while the representations for the largest and the smallest elements of $\mathcal{L}^n$ and the pre-order for $\mathcal{L}^n$ can be trivially obtained from those for $\mathcal{L}$. It follows that we can solve the problem SFM($\mathcal{L}$) in polynomial time in n (in fact, in $n|\mathcal{L}|$, but $|\mathcal{L}|$ is a constant).

To the best of our knowledge, there was up to now not a single non-distributive finite lattice $\mathcal{L}$ for which the problem SFM($\mathcal{L}$) is known to be oracle-tractable. We will provide such examples in this paper.

The problem of minimizing submodular functions on non-distributive lattices was mentioned in [17]. However it was not clear in that paper what the parameter should be in this case, since, in combinatorics, submodular functions are traditionally considered to be defined on (some or all) subsets of a set, and the standard parameter in such situations was always the cardinality of the set. We believe that our formulation of the problem SFM($\mathcal{L}$) is an appropriate form of generalization of the standard SFM problem to the case of arbitrary lattices.

We will generalize this problem even further, by considering classes of finite lattices. Let $\mathbf{C}$ be fixed finite class of finite lattices. Define the optimisation problem SFM($\mathbf{C}$) as follows:

INSTANCE: A lattice $\mathcal{L}' = \mathcal{L}_1 \times \ldots \times \mathcal{L}_n$ such that $\mathcal{L}_i \in \mathbf{C}$ for all $1 \leq i \leq n$, and a supermodular function f on $\mathcal{L}'$.

GOAL: Find an element $\mathbf{a} \in \mathcal{L}'$ such that $f(\mathbf{a}) = \max\{f(\mathbf{b}) \mid \mathbf{b} \in \mathcal{L}'\}$.

One can extend the notion of oracle-tractability to the problems SFM($\mathbf{C}$) in a natural way, assuming that an instance is given by an n-tuple of names of lattices in the product and by an oracle for the function f. For any finite class of finite distributive lattices one can follow the same procedure as for a single distributive lattice, so, clearly, the following statement holds.

Proposition 2.4 SFM($\mathbf{C}$) *is oracle-tractable for any finite class $\mathbf{C}$ of finite distributive lattices.*

3 Maximum constraint satisfaction

Maximum constraint satisfaction problems are well-studied combinatorial optimization problems. The standard example of such problems are MAX k-CUT and MAX k-SAT.

Let D denote a *finite* set with $|D| > 1$. Let $R_D^{(m)}$ denote the set of all m-ary predicates over D, that is, functions from D^m to $\{0,1\}$, and let $R_D = \bigcup_{m=1}^{\infty} R_D^{(m)}$. Also, let $\mathbb{Z}^+$ denote the set of all non-negative integers.

Definition 3.1 *A* constraint *over a set of variables $V = \{x_1, x_2, \ldots, x_n\}$ is an expression of the form $f(\mathbf{x})$ where*

- *$f \in R_D^{(m)}$ is called the* constraint predicate*; and*
- *$\mathbf{x} = (x_{i_1}, \ldots, x_{i_m})$ is called the* constraint scope.

The constraint f is said to be satisfied *on a tuple $\mathbf{a} = (a_{i_1}, \ldots, a_{i_m}) \in D^m$ if $f(\mathbf{a}) = 1$.*

Definition 3.2 *For a finite $\mathcal{F} \subseteq R_D$, an instance of* Max CSP($\mathcal{F}$) *is a pair (V, C) where*

- *$V = \{x_1, \ldots, x_n\}$ is a set of variables taking their values from the set D;*
- *C is a collection of constraints $f_1(\mathbf{x}_1), \ldots, f_q(\mathbf{x}_q)$ over V, where $f_i \in \mathcal{F}$ for all $1 \leq i \leq q$.*

The goal is to find an assignment $\varphi : V \rightarrow D$ that maximizes the number of satisfied constraints, that is, to maximize the function $f : D^n \rightarrow \mathbb{Z}^+$, defined by $f(x_1, \ldots, x_n) = \sum_{i=1}^{q} f_i(\mathbf{x}_i)$. If the constraints have (positive integral) weights ϱ_i, $1 \leq i \leq q$, then the goal is to maximize the total weight of satisfied constraints, to maximize the function $f : D^n \rightarrow \mathbb{Z}^+$, defined by $f(x_1, \ldots, x_n) = \sum_{i=1}^{q} \varrho_i \cdot f_i(\mathbf{x}_i)$.

Example 3.3 [Max k-Cut] In the Max k-Cut problem, one is given an undirected graph $G = (V, E)$ with weighted edges, and the goal is to find a partition of V into k parts, $V = V_0 \cup V_1 \cup \ldots \cup V_{k-1}$, maximizing the total weight of edges with endpoints in different parts. This problem is exactly the Max CSP($\{\neq_k\}$) problem where $\neq_k$ is the binary disequality predicate on $\{0, 1, \ldots, k-1\}$. To see this, think of vertices of a given graph as variables, and apply the predicate to every pair of variables x, y such that (x, y) is an edge in the graph, while keeping all weights the same. $\Box$

Since predicates are functions, one can consider supermodular predicates on a lattice. For a finite lattice $\mathcal{L}$, we will denote by $\mathrm{Spmod}_{\mathcal{L}}$ the set of all predicates that are supermodular on $\mathcal{L}$.

It is easy to see that if $f(x_1, \ldots, x_n) = \sum_{i=1}^{q} \varrho_i \cdot f_i(\mathbf{x}_i)$ and, for some lattice $\mathcal{L}$, every f_i is supermodular on $\mathcal{L}$ then f is also supermodular on $\mathcal{L}$. Moreover, it is clear that one can compute the value of f on a given tuple in linear time in the size of the instance. Hence, we immediately obtain the following lemma.

Lemma 3.4 *Let $\mathcal{L}$ be a finite lattice such that the problem* SFM($\mathcal{L}$) *is oracle-tractable. Then, for any finite set $\mathcal{F} \subseteq \mathrm{Spmod}_{\mathcal{L}}$, the problem* Max CSP($\mathcal{F}$) *is tractable.*

It is intriguing that all known tractable problems Max CSP($\mathcal{F}$) are essentially (i.e., possibly, after removing redundant elements from D) of this form (i.e., with $\mathcal{F} \subseteq \mathrm{Spmod}_{\mathcal{L}}$ for some lattice $\mathcal{L}$ on D). In particular, it is known [3] (and follows from Proposition 2.4 and Lemma 3.4) that Max CSP($\mathcal{F}$) is tractable whenever $\mathcal{F}$ consists of supermodular predicates on some distributive lattice.

In the rest of this section, we present evidence that supermodularity on lattices is probably the right tool for studying the complexity of problems Max CSP($\mathcal{F}$).

First, we will consider a form of supermodular constraints that can be defined on any lattice.

Definition 3.5 *A predicate* $f \in R_D^{(n)}$ *will be called* 2-monotone[1] *on a lattice* $\mathcal{L}$ *on* D *if it can be expressed as follows*

$$f(\mathbf{x}) = 1 \Leftrightarrow ((x_{i_1} \sqsubseteq a_{i_1}) \wedge \ldots \wedge (x_{i_s} \sqsubseteq a_{i_s})) \vee ((x_{j_1} \sqsupseteq b_{j_i}) \wedge \ldots \wedge (x_{j_t} \sqsupseteq b_{j_t})) \quad (1)$$

where $\mathbf{x} = (x_1, \ldots, x_n)$*,* $a_{i_1}, \ldots, a_{i_s}, b_{j_1}, \ldots, b_{j_t} \in D$*, and either of the two disjuncts may be empty (i.e., the value of* s *or* t *may be zero).*

It is straightforward to check that every 2-monotone predicate on a lattice is supermodular on it. The next theorem is, to the best of our knowledge, the only one available on the complexity of supermodular constraints on arbitrary lattices.

Theorem 3.6 ([3]) *Let* $\mathcal{L}$ *be a lattice on a finite set* D*. If* $\mathcal{F}$ *consists of 2-monotone predicates on* $\mathcal{L}$*, then* Max CSP$(\mathcal{F})$ *is tractable.*

Note that 2-monotone predicates can be defined on any poset, since the definition does not use the property of the order to be lattice. However, it was shown in [20] that if $\mathcal{F}$ consists of all binary 2-monotone predicates on a *non-lattice* poset then Max CSP$(\mathcal{F})$ is **NP**-hard.

An *endomorphism* of $\mathcal{F}$ is a unary operation π on D such that, for all $f \in \mathcal{F}$ and all $(a_1, \ldots, a_m) \in D^m$, we have $f(a_1, \ldots, a_m) = 1 \Rightarrow f(\pi(a_1), \ldots, \pi(a_m)) = 1$. We say that $\mathcal{F}$ is a *core* if every endomorphism of $\mathcal{F}$ is injective (i.e. a permutation). The intuition here is that if $\mathcal{F}$ is not a core then it has a non-injective endomorphism π, which implies that, for every assignment φ, there is another assignment $\pi\varphi$ that satisfies all constraints satisfied by φ and uses only a restricted set of values, so the problem Max CSP$(\mathcal{F})$ can be reduced to a similar problem over this smaller set.

Theorem 3.7 ([5, 3, 19]) *Let* $|D| \leq 3$ *and let* $\mathcal{F} \subseteq R_D$ *be a core. If there is a chain* $\mathcal{C}$ *on* D *such that* $\mathcal{F} \subseteq \mathrm{Spmod}_{\mathcal{C}}$ *then* Max CSP$(\mathcal{F})$ *is tractable. Otherwise,* Max CSP$(\mathcal{F})$ *is* **NP***-hard.*

For an element $d \in D$, define the unary predicate u_d so that $u_d(x) = 1 \Leftrightarrow x = d$. Let $C_D = \{u_d \mid d \in D\}$.

Theorem 3.8 ([7, 8]) *Let* D *be any finite set and assume that* $C_D \subseteq \mathcal{F} \subseteq R_D$*. If there is a chain* $\mathcal{C}$ *on* D *such that* $\mathcal{F} \subseteq \mathrm{Spmod}_{\mathcal{C}}$ *then* Max CSP$(\mathcal{F})$ *is tractable. Otherwise,* Max CSP$(\mathcal{F})$ *is* **NP***-hard.*

Note that (assuming that supermodularity is the right tool) chains are the only lattices that could possibly appear in Theorems 3.7 and 3.8 because, as is easy to check, every lattice with at most three elements is a chain and every predicate of the form u_d is supermodular on a lattice if and only if the lattice is a chain (the latter assertion is essentially Lemma 5.1 of [3]).

However, it is known that classes of supermodular predicates on (essentially) different lattices are pairwise incomparable. More precisely, for any lattice $\mathcal{L}$, let $\mathcal{L}^\partial$ denote the dual lattice of $\mathcal{L}$, i.e., the one obtained from $\mathcal{L}$ by reversing the order (or by swapping the lattice operations, which is the same). It is obvious from the definition that the classes of supermodular functions on $\mathcal{L}$ and on $\mathcal{L}^\partial$ coincide. It was shown in [20] that, for any finite lattice $\mathcal{L}'$ (on the same set as $\mathcal{L}$) such that $\mathcal{L}'$ is neither $\mathcal{L}$ nor $\mathcal{L}^\partial$, there exists a predicate which is 2-monotone (and hence supermodular) on $\mathcal{L}$, but not supermodular on $\mathcal{L}'$. Hence, essentially, one cannot exclude *any* lattice from these considerations.

[1] In [3], such predicates are called *generalized* 2-monotone.

It follows that the problem SFM($\mathcal{L}$) restricted to supermodular functions on a non-distributive lattice $\mathcal{L}$, such as the functions that can appear in instances of MAX CSP($\mathcal{F}$), is of special interest.

It is a basic fact in lattice theory (see, e.g., [13]) that a lattice is distributive if it does not contain, as a sublattice, one of the two minimal non-distributive lattices: the pentagon $\mathcal{N}_5$ and the diamond $\mathcal{M}_3$. These two lattices are depicted on Fig. 1.

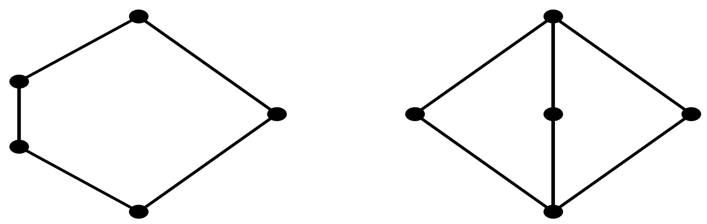

Figure 1: The pentagon $\mathcal{N}_5$ and the diamond $\mathcal{M}_3$.

It will follow from the results in this paper that MAX CSP($\mathcal{F}$) is tractable if $\mathcal{F} \subseteq \text{Spmod}_{\mathcal{N}_5}$ or $\mathcal{F} \subseteq \text{Spmod}_{\mathcal{M}_3}$.

4 SFM and constructions on lattices

In this section we show that tractability of SFM($\mathcal{L}$) is preserved under certain constructions on lattices, and exhibit a large class of non-distributive lattices for which SFM($\mathcal{L}$) is tractable.

4.1 General constructions

A *congruence* on a lattice $\mathcal{L}$ is an equivalence relation θ such that, for all $a, b, c, d \in \mathcal{L}$, the conditions $a\theta b$ and $c\theta d$ imply that both $(a \sqcap c)\theta(b \sqcap d)$ and $(a \sqcup c)\theta(b \sqcup d)$ hold.

If θ is a congruence on $\mathcal{L}$ and $a \in \mathcal{L}$ then let $a[\theta]$ denote the θ-class containing a. It is well-known that every θ-class is a sublattice of $\mathcal{L}$. It is also well known that the family of all θ-classes forms a lattice, called a *factor-lattice* of $\mathcal{L}$ and denoted $\mathcal{L}/\theta$, with operations defined as follows: $a[\theta] \sqcap a'[\theta] = (a \sqcap a')[\theta]$ and $a[\theta] \sqcup a'[\theta] = (a \sqcup a')[\theta]$.

We will now introduce a certain notion of product of classes of lattices that was intensively studied in lattice theory (see, e.g., [14] or pp. 489-90 of [13]).

Definition 4.1 *If* $\mathbf{V}$ *and* $\mathbf{W}$ *are classes of lattices then their* Mal'tsev product*, denoted* $\mathbf{V} \circ \mathbf{W}$*, consists of all lattices* $\mathcal{L}$ *such that there is a congruence* θ *on* $\mathcal{L}$ *with the following properties:*

1. *the lattice* $\mathcal{L}/\theta$ *belongs to* $\mathbf{W}$*;*
2. *every* θ*-class is a lattice from* $\mathbf{V}$*.*

Let $\mathbf{D}$ denote the class of all distributive lattices, let $\mathbf{D}_k$ denote the class of all distributive lattices with at most k elements, and let $\mathbf{D}_{fin}$ denote the class of all finite lattices from $\mathbf{D}$.

Example 4.2 The lattice $\mathcal{N}_5$ belongs to $\mathbf{D}_3 \circ \mathbf{D}_2$. It is easy to check that the equivalence relation θ whose two classes are within the ovals in Fig. 2 is a congruence. The classes of the congruence are distributive lattices (chains), and the lattice $\mathcal{N}_5/\theta$ is a distributive lattice (a two-element chain). $\square$

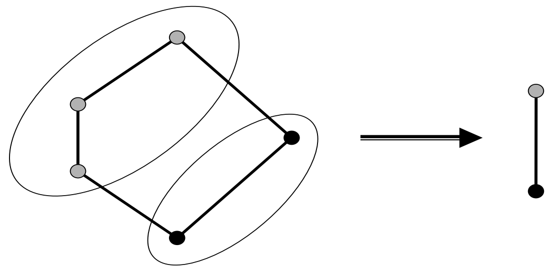

Figure 2: The pentagon $\mathcal{N}_5$ is in $\mathbf{D}_3 \circ \mathbf{D}_2$.

Theorem 4.3 *Suppose that* $\mathbf{V}, \mathbf{W}$ *are finite classes of finite lattices. If* $\mathrm{SFM}(\mathbf{V})$ *and* $\mathrm{SFM}(\mathbf{W})$ *are both oracle-tractable then* $\mathrm{SFM}(\mathbf{V} \circ \mathbf{W})$ *is oracle-tractable as well.*

Proof: Let $\mathcal{L}' = \mathcal{L}_1 \times \ldots \times \mathcal{L}_n$ such that $\mathcal{L}_i \in \mathbf{V} \circ \mathbf{W}$ for all $1 \leq i \leq n$. Then, for every $1 \leq i \leq n$, there exist congruences θ_i such that the lattices $\mathcal{K}_i = \mathcal{L}_i/\theta_i$ belong to $\mathbf{W}$ and every θ_i-class belongs to $\mathbf{V}$.

Define a function f' on $\mathcal{K} = \mathcal{K}_1 \times \ldots \times \mathcal{K}_n$ by letting

$$f'(a_1[\theta_1], \ldots, a_n[\theta_n]) = \max f|_{a_1[\theta_1] \times \ldots \times a_n[\theta_n]}.$$

Let us check that f' is a supermodular function on $\mathcal{K}$. Take two arbitrary elements in $\mathcal{K}$, say $(a_1[\theta_1], \ldots, a_n[\theta_n])$ and $(b_1[\theta_1], \ldots, b_n[\theta_n])$. Choose a'_i, b'_i, $1 \leq i \leq n$, so that $a_i \theta_i a'_i$ and $b_i \theta_i b'_i$ for all $1 \leq i \leq n$, and $f'(a_1[\theta_1], \ldots, a_n[\theta_n]) = f(a'_1, \ldots, a'_n)$ and $f'(b_1[\theta_1], \ldots, b_n[\theta_n]) = f(b'_1, \ldots, b'_n)$. Since each θ_i is a congruence, it follows that

$$f'(a_1[\theta_1], \ldots, a_n[\theta_n]) + f'(b_1[\theta_1], \ldots, b_n[\theta_n]) = f(a'_1, \ldots, a'_n) + f(b'_1, \ldots, b'_n) \leq$$

$$f(a'_1 \sqcap b'_1, \ldots, a'_n \sqcap b'_n) + f(a'_1 \sqcup b'_1, \ldots, a'_n \sqcup b'_n) \leq$$

$$f'((a'_1 \sqcap b'_1)[\theta_1], \ldots, (a'_n \sqcap b'_n)[\theta_n]) + f'((a'_1 \sqcup b'_1)[\theta_1], \ldots, (a'_n \sqcup b'_n)[\theta_n]) =$$

$$f'((a_1 \sqcap b_1)[\theta_1], \ldots, (a_n \sqcap b_n)[\theta_n]) + f'((a_1 \sqcup b_1)[\theta_1], \ldots, (a_n \sqcup b_n)[\theta_n]) =$$

$$f'(a_1[\theta_1] \sqcap b_1[\theta_1], \ldots, a_n[\theta_n] \sqcap b_n[\theta_n]) + f'(a_1[\theta_1] \sqcup b_1[\theta_1], \ldots, a_n[\theta_n] \sqcup b_n[\theta_n])$$

Since $\mathcal{K}$ is a direct product of lattices from $\mathbf{W}$, we infer that f' can be maximized in polynomial time if evaluation of f' on a given tuple is a primitive operation. That is, f' can maximized in at most $p_1(n)$ number of steps, where p_1 is a fixed polynomial, and some of the steps are evaluations of f' on a given element of $\mathcal{K}$.

Assume that $\mathrm{SFM}(\mathbf{V})$ can be solved in $p_2(n)$ steps, some of which are function evaluations. Now, to prove the theorem, it suffices to show that f' can be evaluated on any given element of $\mathcal{K}$ in $p_2(n)$ steps (assuming that evaluating f on a given element of $\mathcal{L}'$ is a primitive operation). Fix an element $(a_1[\theta_1], \ldots, a_n[\theta_n])$ of $\mathcal{K}$. The goal now is to maximize f on $a_1[\theta_1] \times \ldots \times a_n[\theta_n]$. Every $a_i[\theta_i]$ is a lattice from $\mathbf{V}$, so evaluating $f'(a_1[\theta_1], \ldots, a_n[\theta_n])$ can be done in $p_2(n)$ steps by assumption of the theorem. Hence, f can be maximized in $p_1(p_2(n))$ steps, some of which are evaluations of f on a given tuple. $\square$

Corollary 4.4 *If $\mathcal{L}_1$ and $\mathcal{L}_2$ are finite lattices such that* $\mathrm{SFM}(\mathcal{L}_i)$ *is oracle-tractable for $i = 1, 2$, then* $\mathrm{SFM}(\mathcal{L}_1 \times \mathcal{L}_2)$ *is oracle-tractable as well.*

Proof: Let $\mathbf{V} = \{\mathcal{L}_1\}$ and $\mathbf{W} = \{\mathcal{L}_2\}$. It is immediate that $\mathrm{SFM}(\mathbf{V})$ and $\mathrm{SFM}(\mathbf{W})$ are oracle-tractable. The lattice $\mathcal{L}' = \mathcal{L}_1 \times \mathcal{L}_2$ belongs to $\mathbf{V} \circ \mathbf{W}$. Indeed, the relation θ on $\mathcal{L}'$ defined so that $(a_1, b_1)\theta(a_2, b_2)$ if and only if $b_1 = b_2$ is a congruence such that $\mathcal{L}'/\theta$ is isomorphic to $\mathcal{L}_2$ while every θ-class is isomorphic to $\mathcal{L}_1$. Clearly, when maximizing an n-ary supermodular function on $\mathcal{L}_1 \times \mathcal{L}_2$, one can identify $\mathcal{L}'/\theta$ with $\mathcal{L}_2$ and every every θ-class with $\mathcal{L}_1$. The result now follows from Theorem 4.3. $\square$

Definition 4.5 *A mapping φ from a lattice $\mathcal{L}_1$ to a lattice $\mathcal{L}_2$ is called a* homomorphism *if, for all $a, b \in \mathcal{L}_1$, it holds that $\varphi(a \sqcap b) = \varphi(a) \sqcap \varphi(b)$ and $\varphi(a \sqcup b) = \varphi(a) \sqcup \varphi(b)$. If such a mapping φ is onto then $\mathcal{L}_2$ is said to be a* homomorphic image *of $\mathcal{L}_1$.*

Theorem 4.6 *Fix a finite lattice $\mathcal{L}$. If $\mathcal{L}$ is a homomorphic image of some finite lattice $\mathcal{L}_1$ such that* $\mathrm{SFM}(\mathcal{L}_1)$ *is oracle-tractable then* $\mathrm{SFM}(\mathcal{L})$ *is oracle-tractable as well.*

Proof: Every supermodular function f on $\mathcal{L}^n$ can be transformed into a supermodular function f_1 on $\mathcal{L}_1^n$ by letting $f_1(a_1, \ldots, a_n) = f(\varphi(a_1), \ldots, \varphi(a_n))$ where φ is a surjective homomorphism from $\mathcal{L}_1$ to $\mathcal{L}$. It is straightforward to check that f_1 is indeed supermodular on $\mathcal{L}_1$. Since φ is surjective, the maximum of f_1 coincides with the maximum of f. By assumption, one can find a tuple $\mathbf{a}$ maximizing f_1 in polynomial time in n, and the tuple maximizing f is obtained from $\mathbf{a}$ by applying φ component-wise. $\square$

It is clear, say, since the pentagon belongs to $\mathbf{D}_3 \circ \mathbf{D}_2$, that Theorem 4.3 extends (compared to $\mathbf{D}_{fin}$) the class of lattices $\mathcal{L}$ for which $\mathrm{SFM}(\mathcal{L})$ is proved to be oracle-tractable. We note that Theorem 4.6 further extends this class. Consider, for example, the lattice $\hat{\mathcal{L}}$ shown in Fig. 3. It is shown in the proof of Theorem 1 of [14] that this lattice is a homomorphic image of a lattice belonging to $(\mathbf{D}_4 \circ \mathbf{D}_4) \circ \mathbf{D}_4$, so $\mathrm{SFM}(\hat{\mathcal{L}})$ is oracle-tractable. It is easy to verify that this lattice is *simple*, that is, it has no congruences except for the equality relation and the full binary relation. Lemma 1 of [14] states that if a simple lattice belongs to Mal'tsev product of two classes of lattices then it belongs to one of these classes. Hence, since $\hat{\mathcal{L}}$ is simple and non-distributive, it does not belong to any class obtainable from $\mathbf{D}_{fin}$ by using Mal'tsev product.

It follows from Proposition 2.4 and Theorems 4.3 and 4.6 that if a finite lattice $\mathcal{L}$ belongs to a class obtained from finite sets of distributive lattices by (repeatedly) using Mal'tsev product and also taking homomorphic images, then $\mathrm{SFM}(\mathcal{L})$ is oracle-tractable. What is the family $\mathbf{F}$ of finite lattices $\mathcal{L}$ which can be obtained as described above? Unfortunately, it seems quite difficult to give a precise characterisation of this family because, by [22], the process of repeatedly applying Mal'tsev product to $\mathbf{D}$ results in different classes of lattices for different orders of applying the operation. However, in the next subsection we describe a well-understood and rich subclass of $\mathbf{F}$, which can be obtained by repeatedly applying Mal'tsev product only to $\mathbf{D}_2$. The subclass consists of the so-called *bounded* finite lattices.

We will now describe some lattices that definitely do not belong to the family $\mathbf{F}$. For $t \geq 3$, a *t-diamond* (or simply a diamond), denoted $\mathcal{M}_t$, is a lattice on an $(t+2)$-element set such that $0_{\mathcal{M}_t}, 1_{\mathcal{M}_t}$ are the least and the greatest element, respectively, and all t elements in $\mathcal{M}_t \setminus \{0_{\mathcal{M}_t}, 1_{\mathcal{M}_t}\}$

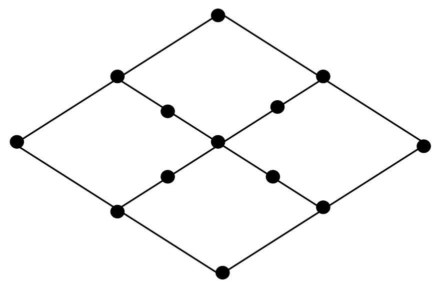

Figure 3: The lattice $\hat{\mathcal{L}}$.

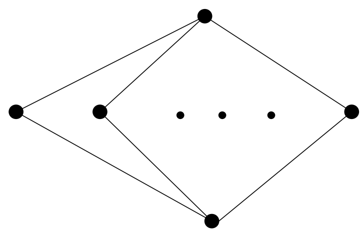

Figure 4: A diamond lattice $\mathcal{M}_t$.

are pairwise incomparable. The Hasse diagram of $\mathcal{M}_t$ is given in Fig. 4. Note that $\mathcal{M}_3$ is often referred to as *the* diamond. It is well-known and easy to check that every diamond is a simple non-distributive lattice. In addition, every diamond has the property that, for every finite lattice $\mathcal{L}$ having $\mathcal{M}_t$ as a homomorphic image, $\mathcal{L}$ also contains $\mathcal{M}_t$ as a sublattice (see Lemma 6.21 of [15] or the more general Theorem 2.47 of [18]). Hence, in order to show that no diamond belongs to $\mathbf{F}$, it is sufficient to prove that if a finite lattice $\mathcal{L}$ containing $\mathcal{M}_t$ as a sublattice belongs to $\mathbf{X} \circ \mathbf{Y}$ then at least one of these two classes contains a lattice with the same property. So assume that $\mathcal{L} \in \mathbf{X} \circ \mathbf{Y}$. By definition, $\mathcal{L}$ has a congruence θ such that $\mathcal{L}/\theta \in \mathbf{Y}$ and every θ-class is in $\mathbf{X}$. It is well-known, and easy to show that the restriction θ' of θ on $\mathcal{M}_t$ is a congruence of $\mathcal{M}_t$. By simplicity of $\mathcal{M}_t$, all the elements of $\mathcal{M}_t$ belong either to a single θ'-class or to pairwise different θ'-classes. In the former case, $\mathcal{M}_t$ is entirely contained in some θ-class, so this class is the required lattice in $\mathbf{X}$. In the latter case, $\mathcal{L}/\theta \in \mathbf{Y}$ is the required lattice. This argument can be easily generalized to show that, in fact, no lattice containing $\mathcal{M}_3$ as a sublattice belongs to $\mathbf{F}$.

Even though we have been unable to prove that $\mathrm{SFM}(\mathcal{M}_t)$ is oracle-tractable, we will show in Section 5 that the problems MAX CSP($\mathcal{F}$) with $\mathcal{F} \subseteq \mathrm{Spmod}_{\mathcal{M}_t}$ are tractable.

4.2 Finite bounded lattices

Let $\mathcal{L}$ be a lattice and $u \sqsubseteq v$ two comparable (and not necessarily distinct) elements of $\mathcal{L}$. Let I denote the interval $[u, v] = \{x \in \mathcal{L} \mid u \sqsubseteq x \sqsubseteq v\}$ in $\mathcal{L}$.

Definition 4.7 *The lattice $\mathcal{L}[I]$ is said to be obtained from $\mathcal{L}$ by* doubling the interval I *if the base set of $\mathcal{L}[I]$ is $(\mathcal{L} \setminus I) \cup (I \times \{0, 1\})$ and $x \sqsubseteq y$ holds in $\mathcal{L}[I]$ if and only if*

- *$x \sqsubseteq y$ in $\mathcal{L}$ and $x, y \notin I$, or*

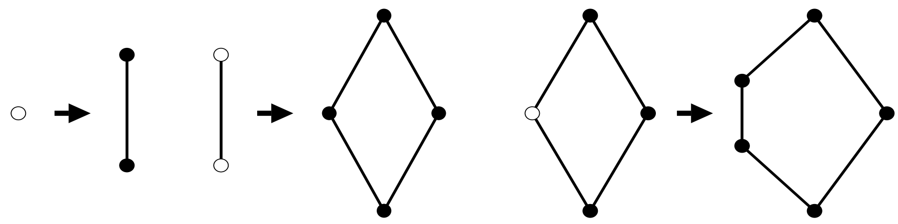

Figure 5: Obtaining the pentagon by doubling intervals (in three steps). In each step, the white elements indicate the interval to be doubled.

- *$x \in \mathcal{L} \setminus I$, $y = (b, j)$ and $x \sqsubseteq b$ in $\mathcal{L}$, or*
- *$x = (a, i)$, $y \in \mathcal{L} \setminus I$ and $a \sqsubseteq y$ in $\mathcal{L}$, or*
- *$x = (a, i)$, $y = (b, j)$ such that $a, b \in I$ and $a \sqsubseteq b$ in $\mathcal{L}$, and $i \leq j$.*

Essentially, doubling an interval I in $\mathcal{L}$ means replacing it with $I \times \{0, 1\}$.

Definition 4.8 *A finite lattice $\mathcal{L}$ is called* bounded[2] *if there is a sequence $\mathcal{L}_1, \ldots, \mathcal{L}_n$ such that $\mathcal{L}_1$ is one-element, $\mathcal{L}_n$ is isomorphic to $\mathcal{L}$, and, for $j = 1, \ldots, n-1$, there exists an interval I_j in $\mathcal{L}_j$ such that $\mathcal{L}_{j+1}$ is isomorphic to $\mathcal{L}_j[I_j]$.*

In other words, a finite lattice is called bounded if it can be obtained from the one-element lattice by successive doubling of intervals. For example, Fig. 5 demonstrates that the pentagon is a bounded lattice, while it is not hard to check the lattice $\hat{\mathcal{L}}$ from Fig. 3 is not bounded. Bounded lattices play an important role in lattice theory [18]. The name "bounded" comes from an equivalent characterisation of such lattices, which is usually used as the definition (but does not play any role in this paper). Theorem 2.46 of [18] states that a finite lattice $\mathcal{L}$ is bounded if and only if there exists a free lattice $\mathcal{FL}$ and a congruence α on $\mathcal{FL}$ such that $\mathcal{L}$ is isomorphic to $\mathcal{FL}/\alpha$ and every α-class in $\mathcal{FL}$ has both the least and the greatest element.

It is easy to check that for any lattice $\mathcal{L}$ and any interval I in $\mathcal{L}$, the binary relation θ on $\mathcal{L}[I]$ defined by the rule

$$(x, y) \in \theta \Leftrightarrow \text{ either } x = y \notin I \text{ or } x = (a, i), y = (a, j) \text{ for some } a \in I$$

is a congruence of $\mathcal{L}[I]$. Moreover, the lattice $\mathcal{L}[I]/\theta$ is isomorphic to $\mathcal{L}$, and every θ-class is either a one-element lattice $\mathcal{C}_1$ or a two-element chain $\mathcal{C}_2$. Therefore, we have that $\mathcal{L}[I]$ belongs to the Mal'tsev product $\{\mathcal{C}_1, \mathcal{C}_2\} \circ \{\mathcal{L}\}$. So, if we inductively define $\mathbf{D}_2^1 = \mathbf{D}_2$ and $\mathbf{D}_2^n = \mathbf{D}_2 \circ \mathbf{D}_2^{n-1}$ for $n > 1$, then we have that every finite bounded lattice belongs to $\mathbf{D}_2^n$ for a suitable n. Hence, Theorem 4.3 (together with Lemma 3.4) implies the following statement.

Proposition 4.9 *Let $\mathcal{L}$ be a fixed finite bounded lattice. Then* SFM($\mathcal{L}$) *is oracle-tractable and, for any finite set $\mathcal{F} \subseteq \text{Spmod}_{\mathcal{L}}$,* Max CSP($\mathcal{F}$) *is tractable.*

It is known (see Lemma 2.40 of [18]) that the class of finite bounded lattices is a pseudo-variety, that is, it is closed under taking homomorphic images, sublattices, and finite direct products. Since

[2] Not to be confused with posets that have both the least and the greatest element.

the two-element chain is a bounded lattice, it follows that every finite distributive lattice is bounded. Interestingly, if one allows not only finite, but arbitrary direct products (i.e., direct products of infinitely many lattices) then one can generate *all* lattices from finite bounded lattices (this fact immediately follows from Theorems 2.25 and 2.44 of [18]). In other words, every (not necessarily finite) lattice is a homomorphic image of a subalgebra of a direct product of (possibly infinitely many) finite bounded lattices. Unfortunately, this fact seems to be useless for algorithmic aspects which we are interested in.

5 Max CSP on diamonds

In this section, we consider problems Max CSP($\mathcal{F}$) where $\mathcal{F}$ consists of supermodular predicates on a diamond $\mathcal{M}_t$ (see Fig. 4). The middle elements of $\mathcal{M}_t$ are called *atoms*. Note that, for every pair of distinct atoms a and b, we have $a \sqcap b = 0_{\mathcal{M}_t}$ and $a \sqcup b = 1_{\mathcal{M}_t}$. For simplicity, let $\mathcal{L}$ denote an arbitrary (fixed) t-diamond, $t \geq 3$, throughout this section.

5.1 The structure of supermodular predicates on diamonds

In this subsection, we describe the structure of supermodular predicates on $\mathcal{L}$ by representing them as logical formulas involving constants (elements of $\mathcal{L}$) and the order relation $\sqsubseteq$ of $\mathcal{L}$.

For a subset $D' \subseteq D$, let $u_{D'}$ denote the predicate such that $u_{D'}(x) = 1 \Leftrightarrow x \in D'$. The following lemma can be easily derived directly from the definition of supermodularity.

Lemma 5.1 *A unary predicate $u_{D'}$ is in* $\mathrm{Spmod}_{\mathcal{L}}$ *if and only if either both $0_{\mathcal{L}}, 1_{\mathcal{L}} \in D'$ or else $|D'| \leq 2$ and at least one of $0_{\mathcal{L}}, 1_{\mathcal{L}}$ is in D'.*

For a sequence $\mathbf{y} = (x_{i_1}, \ldots, x_{i_m})$ of variables and a sequence $\mathbf{c} = (c_{i_1}, \ldots, c_{i_m})$ of elements of $\mathcal{L}$, we write $\mathbf{y} \sqsubseteq \mathbf{c}$ to denote $\bigwedge_{1 \leq s \leq m} (x_{i_s} \sqsubseteq c_{i_s})$, and the condition $\mathbf{y} \sqsupseteq \mathbf{c}$ is defined dually.

Theorem 5.2 *Every predicate $f(x_1, \ldots, x_n)$ in* $\mathrm{Spmod}_{\mathcal{L}}$*, such that f takes both values 0 and 1, can be represented as one of the following logical implications:*

1. $[(x_i \sqsubseteq a_1) \vee \ldots \vee (x_i \sqsubseteq a_l)] \implies (x_i \sqsubseteq 0_{\mathcal{L}})$ *where the a_j's are atoms;*
2. $\neg(\mathbf{y} \sqsupseteq \mathbf{c}) \implies (\mathbf{z} \sqsubseteq \mathbf{d})$ *where $\mathbf{y}$ and $\mathbf{z}$ are some subsequences of $(x_1, \ldots, x_n)$, and $\mathbf{c}$, $\mathbf{d}$ are tuples of elements of $\mathcal{L}$ (of corresponding length) such that $\mathbf{c}$ contains no $0_{\mathcal{L}}$ and $\mathbf{d}$ no $1_{\mathcal{L}}$;*
3. $[(x_i \sqsubseteq b_1) \vee \cdots \vee (x_i \sqsubseteq b_k) \vee \neg(\mathbf{y} \sqsupseteq \mathbf{c})] \implies (x_i \sqsubseteq a)$ *where the b_j's are atoms, $\mathbf{y}$ does not contain x_i, and $a \neq 1_{\mathcal{L}}$;*
4. $\neg(x_i \sqsupseteq b) \implies [\neg(x_i \sqsupseteq a_1) \wedge \cdots \wedge \neg(x_i \sqsupseteq a_l) \wedge (\mathbf{y} \sqsubseteq \mathbf{c})]$ *where the a_j's are atoms, $\mathbf{y}$ does not contain x_i, and $b \neq 0_{\mathcal{L}}$;*
5. $\neg(\mathbf{y} \sqsupseteq \mathbf{c}) \implies$ ***false*** *where $\mathbf{y}$ is a subsequence of $(x_1, \ldots, x_n)$ and $\mathbf{c}$ contains no $0_{\mathcal{L}}$;*
6. ***true*** $\implies (\mathbf{y} \sqsubseteq \mathbf{c})$ *where $\mathbf{y}$ is a subsequence of $(x_1, \ldots, x_n)$ and $\mathbf{c}$ contains no $1_{\mathcal{L}}$.*

Conversely, every predicate that can be represented in one of the above forms belongs to $\mathrm{Spmod}_{\mathcal{L}}$.

Example 5.3 The unary predicate of type (1) above is the same as $u_{D'}$ where $D' = D \backslash \{a_1, \ldots, a_l\}$. The predicates $u_{D'} \in \mathrm{Spmod}_{\mathcal{L}}$ with $|D'| \leq 2$ are the unary predicates of types (5) and (6). $\square$

Remark 5.4 Note that constraints of types (2),(5), and (6) are 2-monotone on $\mathcal{L}$, while constraints of types (3) and (4) (and most of those of type (1)) are not.

Proof: (of Theorem 5.2). It is straightforward to verify that all the predicates in the list are actually supermodular. Now we prove the converse. Consider first the case where the predicate f is *essentially unary*, i.e., there is a variable x_i such that $f(x_1, \ldots, x_n) = u_{D'}(x_i)$ for some $D' \subsetneq D$. If $D' = \{x : x \sqsubseteq a\}$ or $D' = \{x : x \sqsupseteq a\}$ for some atom a then f is of the form (5) or (6); otherwise both $0_{\mathcal{L}}$ and $1_{\mathcal{L}}$ are in D' by Lemma 5.1, and if $a_1, \ldots, a_l$ denote the atoms of the lattice that are not in D', then it is clear that f is described by the implication (1).

Now we may assume that f is not essentially unary. If it is 2-monotone, then it is easy to see that f must be described by an implication of type (2), (5) or (6). So now we assume that f is not essentially unary and it is not 2-monotone; we prove that it is described by an implication of type (3) or (4). We require a few claims:

Claim 0. *The set X of all tuples $\mathbf{u}$ such that $f(\mathbf{u}) = 1$ is a sublattice of $\mathcal{L}^n$, i.e. is closed under join and meet.*

This follows immediately from the supermodularity of f.

Claim 1. *There exist indices $1 \leq i_1, \ldots, i_k, j_1, \ldots, j_l \leq n$, atoms $e_1, \ldots, e_k$ and $b_1, \ldots, b_l$ of $\mathcal{L}$ such that $f(\mathbf{x}) = 1$ if and only if*

$$[\neg(x_{i_1} \sqsupseteq e_1) \wedge \cdots \wedge \neg(x_{i_k} \sqsupseteq e_k)] \bigvee [\neg(x_{j_1} \sqsubseteq b_1) \wedge \cdots \wedge \neg(x_{j_l} \sqsubseteq b_l)].$$

Notice first that the set Z of tuples $\mathbf{u}$ such that $f(\mathbf{u}) = 0$ is *convex* in $\mathcal{L}^n$, i.e. if $\mathbf{u} \sqsubseteq \mathbf{v} \sqsubseteq \mathbf{w}$ with $f(\mathbf{u}) = f(\mathbf{w}) = 0$ then $f(\mathbf{v}) = 0$. To show this we construct a tuple $\mathbf{v}'$ as follows: for each coordinate i it is easy to find an element v'_i such that $v_i \sqcap v'_i = u_i$ and $v_i \sqcup v'_i = w_i$. Hence $\mathbf{v} \sqcap \mathbf{v}' = \mathbf{u}$ and $\mathbf{v} \sqcup \mathbf{v}' = \mathbf{w}$ so by supermodularity of f neither $\mathbf{v}$ nor $\mathbf{v}'$ is in X. It follows in particular that neither $0_{\mathcal{L}^n}$ nor $1_{\mathcal{L}^n}$ is in Z; indeed, if $0_{\mathcal{L}^n} \in Z$, let $\mathbf{a}$ be the smallest element in X (the meet of all elements in X), which exists by Claim 0. Since Z is convex it follows that every element above $\mathbf{a}$ is in X so f is 2-monotone, a contradiction. The argument for $1_{\mathcal{L}^n}$ is identical.

Now let $\mathbf{w} \in Z$ be minimal, and let $\mathbf{v} \sqsubseteq \mathbf{w}$. As above we can find a tuple $\mathbf{v}'$ such that $\mathbf{v} \sqcup \mathbf{v}' = \mathbf{w}$ and $\mathbf{v} \sqcap \mathbf{v}' = 0_{\mathcal{L}^n}$; by supermodularity of f it follows that $\mathbf{v} = \mathbf{w}$ or $\mathbf{v}' = \mathbf{w}$. It is easy to deduce from this that there exists a coordinate s such that w_s is an atom of $\mathcal{L}$ and $w_t = 0_{\mathcal{L}}$ for all $t \neq s$. A similar argument shows that every maximal element of Z has a unique coordinate which is an atom and all others are equal to $1_{\mathcal{L}}$.

Since Z is convex, we have that $f(\mathbf{x}) = 0$ if and only if $\mathbf{x}$ is above some minimal element of Z and below some maximal element of Z; Claim 1 then follows immediately.

For each index $i \in \{i_1, \ldots, i_k\}$ that appears in the expression in Claim 1, there is a corresponding condition of the form

$$\neg(x_i \sqsupseteq e_{s_1}) \wedge \cdots \wedge \neg(x_i \sqsupseteq e_{s_r});$$

let I_i denote the set of elements of $\mathcal{L}$ that satisfy this condition. Obviously it cannot contain $1_{\mathcal{L}}$ and must contain $0_{\mathcal{L}}$. Similarly, define for each index $j \in \{j_1, \ldots, j_l\}$ the set F_j of all elements of $\mathcal{L}$ that satisfy the corresponding condition of the form

$$\neg(x_j \sqsubseteq b_{t_1}) \wedge \cdots \wedge \neg(x_j \sqsubseteq b_{t_q});$$

it is clear that $0_{\mathcal{L}} \notin F_j$ and $1_{\mathcal{L}} \in F_j$.

The condition of Claim 1 can now be rephrased as follows: $f(\mathbf{x}) = 1$ *if and only if* $x_i \in I_i$ *for all* $i \in \{i_1, \ldots, i_k\}$ *or* $x_j \in F_j$ *for all* $j \in \{j_1, \ldots, j_l\}$. It is straightforward to verify that since f is not 2-monotone, one of the I_i or one of the F_j must contain 2 distinct atoms. We consider the first case, and we show that the predicate f is of type (4). The case where some F_j contains two atoms is dual and will yield type (3).

Claim 2. *Suppose that I_i contains distinct atoms c and d for some $i \in \{i_1, \ldots, i_k\}$. Then (a) i is the only index with this property, (b) $\{j_1, \ldots, j_l\} = \{i\}$ and (c) F_i does not contain 2 distinct atoms.*

We prove (b) first. We have that

$$f(0_{\mathcal{L}}, \ldots, 0_{\mathcal{L}}, c, 0_{\mathcal{L}}, \ldots, 0_{\mathcal{L}}) = f(0_{\mathcal{L}}, \ldots, 0_{\mathcal{L}}, d, 0_{\mathcal{L}}, \ldots, 0_{\mathcal{L}}) = 1$$

(where c and d appear in the i-th position) and by supermodularity it follows that we also have $f(0_{\mathcal{L}}, \ldots, 0_{\mathcal{L}}, 1_{\mathcal{L}}, 0_{\mathcal{L}}, \ldots, 0_{\mathcal{L}}) = 1$. Since I_i does not contain $1_{\mathcal{L}}$, we have that $x_j \in F_j$ for each $j \in \{j_1, \ldots, j_l\}$; since F_j never contains $0_{\mathcal{L}}$, (b) follows immediately. Since $\{j_1, \ldots, j_l\}$ is non-empty, (a) follows immediately from (b). Finally, if F_i contained distinct atoms then by dualising the preceding argument we would obtain that $\{i_1, \ldots, i_k\} = \{i\}$ from which it would follow that f would be essentially unary, contrary to our assumption. This concludes the proof of the claim.

Let b denote the minimal element in F_i, and for each index $s \in \{i_1, \ldots, i_k\}$ different from i let c_s denote the (unique) maximal element of I_s; then we can describe f as follows: $f(\mathbf{x}) = 1$ if and only if

$$[x_i \in I_i \wedge (\mathbf{y} \sqsubseteq \mathbf{c})] \vee (x_i \sqsupseteq b)$$

where $\mathbf{y}$ is a tuple of variables different from x_i and $\mathbf{c}$ is the tuple whose entries are the c_s defined previously. It remains to rewrite the condition $x_i \in I_i$. Suppose first that there exists at least one atom of $\mathcal{L}$ outside I_i, and let $a_1, \ldots, a_l$ denote the atoms outside I_i. Then it is clear that $x_i \in I_i$ if and only if $\neg(x_i \sqsupseteq a_1) \vee \cdots \vee \neg(x_i \sqsupseteq a_l)$ holds, so the predicate f is of type (4) (simply restate the disjunction as an implication). Now for the last possibility, where I_i contains all of D except $1_{\mathcal{L}}$; then it is easy to see that f can be described by the following:

$$[\neg(x_i \sqsupseteq b) \wedge (\mathbf{y} \sqsubseteq \mathbf{c})] \vee (x_i \sqsupseteq b)$$

and this completes the proof of the theorem. □

We remark that the preceding theorem can be extended to give a similar characterisation of the supermodular constraints on lattices in the larger class of so-called *relatively complemented lattices* [21].

5.2 Supermodular constraints on diamonds are tractable

Theorem 5.5 *If $\mathcal{F} \subseteq \mathrm{Spmod}_{\mathcal{L}}$ then* Max CSP($\mathcal{F}$) *can be solved (to optimality) in $O(n^3 \cdot |\mathcal{L}|^3 + q^3)$ time where n is the number of variables and q is the number of constraints in an instance.*

Proof: We will show how the problem can be reduced to the well-known tractable problem Min Cut.

Let $\mathcal{I} = \{\rho_1 \cdot f_1(\mathbf{x}_1), \ldots, \rho_q \cdot f_q(\mathbf{x}_q)\}$, $q \geq 1$, be an instance of weighted Max CSP($\mathcal{F}$), over a set of variables $V = \{x_1, , \ldots, , x_n\}$.

Construction.

Let ∞ denote an integer greater than $\sum \rho_i$. For each constraint f_i, fix a representation as described in Theorem 5.2. In the following construction, we will refer to the type of f_i which will be a number from 1 to 6 according to the type of representation. Every condition of the form

$(\mathbf{y} \sqsubseteq \mathbf{c})$ will be read as $\bigwedge(x_{i_s} \sqsubseteq c_{i_s})$, and every condition of the form $\neg(\mathbf{y} \sqsupseteq \mathbf{c})$ as $\bigvee \neg(x_{i_s} \sqsupseteq c_{i_s})$, where i_s runs through the indices of variables in $\mathbf{y}$. Moreover, we replace every (sub)formula of the form $\neg(x \sqsupseteq 1_{\mathcal{L}})$ by $\bigvee_{i=1}^{n} \neg(x \sqsupseteq a_i)$ where $a_1, \ldots, a_n$ are the atoms of $\mathcal{L}$.

We construct a digraph $G_{\mathcal{I}}$ as follows:

- The vertices of $G_{\mathcal{I}}$ are as follows
  - $\{T, F\} \cup \{x_d \mid x \in V, d \in \mathcal{L}\} \cup \{\bar{x}_d \mid x \in V, d \in \mathcal{L} \text{ is an atom}\} \cup \{e_i, \bar{e}_i \mid i = 1, 2, \ldots, q\}$[3].

  For each f_i of type (5), we identify the vertex e_i with F. Similarly, for each f_i of type (6), we identify the vertex $\bar{e}_i$ with T.

- The arcs of $G_{\mathcal{I}}$ are defined as follows:
  - For each atom c in $\mathcal{L}$ and for each $x \in V$, there is an arc from $x_{0_{\mathcal{L}}}$ to x_c with weight ∞, and an arc from $\bar{x}_c$ to $x_{1_{\mathcal{L}}}$ with weight ∞;
  - For each pair of distinct atoms c, d in $\mathcal{L}$ and for each $x \in V$, there is an arc from x_c to $\bar{x}_d$ with weight ∞;
  - For each f_i, there is an arc from $\bar{e}_i$ to e_i with weight ρ_i;
  - For each f_i of types (1-4), and each subformula of the form $(x \sqsubseteq a)$ or $\neg(x \sqsupseteq a)$ in the consequent of f_i, there is an arc from e_i to x_a or $\bar{x}_a$, respectively, with weight ∞;
  - For each f_i of types (1-4), and each subformula of the form $(x \sqsubseteq a)$ or $\neg(x \sqsupseteq a)$ in the antecedent of f_i, there is an arc from x_a or $\bar{x}_a$, respectively, to $\bar{e}_i$, with weight ∞;
  - For each f_i of type (5), and each subformula of the form $\neg(x \sqsupseteq a)$ in it, there is an arc from $\bar{x}_a$ to $\bar{e}_i$ with weight ∞;
  - For each f_i of type (6), and each subformula of the form $(x \sqsubseteq a)$ in it, there is an arc from e_i to x_a with weight ∞;

Arcs with weight less than ∞ will be called *constraint* arcs.

It is easy to see that $G_{\mathcal{I}}$ is a digraph with source T (corresponding to **true**) and sink F (corresponding to **false**). Note that paths of non-constraint arcs between vertices corresponding to any given variable $x \in V$ precisely correspond to logical implications that hold between the corresponding assertions. Throughout the proof, we say "a cut in $G_{\mathcal{I}}$" meaning a (T, F)-cut.

Define the *deficiency* of an assignment φ as the difference between $\sum_{i=1}^{q} \rho_i$ and the evaluation of φ on $\mathcal{I}$. In other words, the deficiency of φ is the total weight of constraints not satisfied by φ. We will prove that minimal cuts in $G_{\mathcal{I}}$ exactly correspond to optimal assignments to $\mathcal{I}$. More precisely, we will show that, for each minimal cut in $G_{\mathcal{I}}$ with weight ρ, there is an assignment for $\mathcal{I}$ with deficiency at most ρ, and, for each assignment to $\mathcal{I}$ with deficiency ρ', there is a cut in $G_{\mathcal{I}}$ with weight ρ'.

The semantics of the construction of $G_{\mathcal{I}}$ will be as follows: the vertices of the form x_a or $\bar{x}_a$ correspond to assertions of the form $x \sqsubseteq a$ or $\neg(x \sqsupseteq a)$, respectively, and arcs denote implications about these assertions. Given a minimal cut in $G_{\mathcal{I}}$, we will call a vertex x_a *reaching* if F can be reached from it without crossing the cut. Furthermore, if a vertex x_a is reaching then this will designate that the corresponding assertion is false, and otherwise the corresponding assertion is true. A constraint is not satisfied if and only if the corresponding constraint arc crosses the cut.

[3]The vertices x_d will correspond to the expressions $x \sqsubseteq d$ and $\bar{x}_d$ to $\neg(x \sqsupseteq d)$.

Let C be a minimal cut in $G_{\mathcal{I}}$. Obviously, C contains only constraint arcs. First we show that, for every variable $x \in V$, there is a unique minimal element $a \in \mathcal{L}$ such that x_a is non-reaching. All we need to show is the following: if c, d are distinct atoms such that both x_c and x_d are both non-reaching then so is $x_{0_{\mathcal{L}}}$. Assume that, on the contrary, $x_{0_{\mathcal{L}}}$ is reaching. Then there is a path from $x_{0_{\mathcal{L}}}$ to F not crossing the cut. It is easy to notice that such a path has to go through a vertex $\bar{x}_a$ for some atom $a \in \mathcal{L}$, since the second vertex on this path must be of the form x_b for some atom b, and it is followed either by a vertex $\bar{x}_a$ or else by three vertices $\bar{e}_i, e_i, x_{b'}$ for some $1 \leq i \leq q$ and some atom b'. However, we have an arc from at least one of vertices x_c, x_d to $\bar{x}_a$, and hence at least one of these vertices would have a path to F not crossing the cut, a contradiction.

Note that, for every $x \in V$, there are no arcs coming out of $x_{1_{\mathcal{L}}}$. Hence, for every $x \in V$, there is a unique minimal element $v \in \mathcal{L}$ such that F cannot be reached from x_v without crossing the cut.

Define an assignment φ_C as follows:

$$\varphi_C(x) \text{ is the unique minimal element } a \text{ such that } x_a \text{ is non-reaching.}$$

We now make some observations. Note that, for all $x \in V$ and $a \in \mathcal{L}$, we have that $\varphi_C(x) \sqsubseteq a$ if and only x_a is non-reaching. Moreover, if $\bar{x}_a$ is reaching then, for each atom $b \neq a$, we have an arc from x_b to $\bar{x}_a$ meaning that $\varphi_C(x) \not\sqsubseteq b$, and hence $\varphi_C(x) \sqsupseteq a$. Furthermore, if $\bar{x}_a$ is non-reaching then $\varphi_C(x) \neq a$. Indeed, if $\varphi_C(x) = a$ then x_b is reaching for all atoms $b \neq a$, and, since every path from x_b to F has to go through a vertex $\bar{x}_c$ for some c, we have that $\bar{x}_c$ is reaching. Then $c \neq a$, and there is an arc from x_a to $\bar{x}_c$, so x_a is reaching, a contradiction. To summarize,

- if a node of the form x_a or $\bar{x}_a$ is reaching then the corresponding assertion is falsified by the assignment φ_C;
- if a node of the form x_a is non-reaching then $\varphi_C(x) \sqsubseteq a$;
- if a node of the form $\bar{x}_a$ is non-reaching then the truth value of the corresponding assertion is undecided.

Suppose that a constraint arc corresponding to a constraint f_i is not in the cut. We claim that f_i is satisfied by the assignment φ_C. To show this, we will go through the possible types of f_i.

If f_i is of type (1), (2), (5), or (6), then the claim is straightforward. For example, let f_i be of type (1). If the node $x_{0_{\mathcal{L}}}$ corresponding to the consequent is reaching, then so are all nodes corresponding to the antecedent. Hence, all atomic formulas are falsified by the assignment φ_C, and the implication is true. If $x_{0_{\mathcal{L}}}$ is non-reaching, then $\varphi_C(x) = 0_{\mathcal{L}}$, and the constraint is clearly satisfied. The argument for types (2), (5), (6) is very similar.

Let f_i be of type (3). Then, if the node corresponding to the consequent is non-reaching then the consequent is satisfied by φ_C, and so the constraint is satisfied. If this node is reaching then every node corresponding to the disjuncts in the antecedent is reaching. Then both antecedent and consequent are falsified by φ_C, and the constraint is satisfied.

Let f_i be of type (4), that is, of the form

$$\neg(x_i \sqsupseteq b) \implies [\neg(x_i \sqsupseteq a_1) \wedge \cdots \wedge \neg(x_i \sqsupseteq a_l) \wedge (\mathbf{y} \sqsubseteq \mathbf{c})].$$

If a node corresponding to some conjunct in the consequent is reaching, then the node corresponding to the antecedent is also reaching. So $\varphi_C(x_i) \sqsupseteq b$, and the constraint is satisfied. More generally, if the node corresponding to the antecedent is reaching then the constraint is satisfied regardless of what happens with the consequent. Assume that all nodes corresponding to conjuncts in the consequent and in the antecedent are non-reaching. Then the conjunct $(\mathbf{y} \sqsubseteq \mathbf{c})$ is satisfied by φ_C. Furthermore, we know (see the observations above) that $\varphi_C(x_i) \neq b$, and also that $\varphi_C(x_i) \neq a_s$ for $1 \leq s \leq l$. If $\varphi_C(x_i) = 1_{\mathcal{L}}$ then both the antecedent and the consequent of f_i are false, and hence f_i is satisfied. Otherwise, $\varphi_C(x_i) \not\sqsupseteq b$ and $\varphi_C(x_i) \not\sqsupseteq a_s$ for $1 \leq s \leq l$, so f_i is satisfied anyway.

Conversely, let φ be an assignment to $\mathcal{I}$, and let K be the set of constraints in $\mathcal{I}$ that are not satisfied by φ. Consider any path from T to F. It is clear that if all constraints corresponding to constraint arcs on this path are satisfied, then we have a chain of valid implications starting from **true** and finishing at **false**. Since this is impossible, at least one constraint corresponding to such an arc is not satisfied by φ. Hence, the constraint arcs corresponding to constraints in K form a cut in $G_{\mathcal{I}}$. Furthermore, by the choice of K, the weight of this cut is equal to the deficiency of φ.

It follows that the standard algorithm [12] for the Min Cut problem can be used to find an optimal assignment for any instance of Max CSP($\mathcal{F}$). This algorithm runs in $O(k^3)$ where k is the number of vertices in the graph. Since the number of vertices in $G_{\mathcal{I}}$ is at most $2(1 + n \cdot |D| + q)$, the result follows. □

We remark that a partial converse to Theorem 5.5 was proved in [20] where it is shown that if $\mathcal{F}$ contains all 2-monotone predicates on a diamond and any predicate which is not supermodular on that diamond then Max CSP($\mathcal{F}$) is **NP**-hard.

6 Conclusion

We have described a large class of lattices on which the SFM problem is oracle-tractable, and an even larger class of lattices $\mathcal{L}$ such that Max CSP($\mathcal{F}$) is tractable whenever all predicates in $\mathcal{F}$ are supermodular on $\mathcal{L}$. We believe that more progress in the study of maximum constraint satisfaction can be achieved by further blending supermodular optimization and algebraic lattice theory.

The three most standard constructions on algebras (and on lattices in particular) are the forming of homomorphic images, subalgebras, and direct products. We showed (Corollary 4.4 and Theorem 4.6) that two of them preserve oracle-tractability of the problems SFM($\mathcal{L}$). However, it is an open problem whether, for any finite lattice $\mathcal{L}_1$, oracle-tractability of SFM($\mathcal{L}_1$) implies oracle-tractability of SFM($\mathcal{L}_2$) for any sublattice $\mathcal{L}_2$ of $\mathcal{L}_1$.

This paper explores two ways of obtaining tractability results for Max CSP problems – one is via oracle-tractability of the SFM problem, and the other via explicit description of supermodular predicates. One other way that remains to be explored has to do with implicit methods (see [2, 4]) of showing that some predicates can be simulated (or "strictly implemented" [8, 19]) by other predicates. One interesting question in this connection is whether any supermodular predicate on a lattice can be simulated, i.e., strictly implemented, by 2-monotone predicates on that lattice. A positive answer to this question would, together with Theorem 3.6, imply that Max CSP($\mathcal{F}$) is tractable for arbitrary sets $\mathcal{F}$ of supermodular predicates on a lattice.

Finally, we hope that the study of sub- and supermodular functions on lattices will find more applications in combinatorial optimization.

References

[1] R.E. Burkard, B. Klinz, and R. Rudolf. Perspectives of Monge properties in optimization. *Discrete Applied Mathematics*, 70:95–161, 1996.

[2] D. Cohen, M. Cooper, and P. Jeavons. An algebraic characterisation of complexity for valued constraints. In *CP'06*, volume 4204 of *LNCS*, pages 107–121, 2006.

[3] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin. Supermodular functions and the complexity of Max CSP. *Discrete Applied Mathematics*, 149(1-3):53–72, 2005.

[4] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin. The complexity of soft constraint satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.

[5] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications*. 2001.

[6] R. Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[7] V. Deineko, P. Jonsson, M. Klasson, and A. Krokhin. Supermodularity on chains and complexity of maximum constraint satisfaction. In *Proceedings of Eurocomb'05, DMTCS*, volume AE, pages 51–56, 2005.

[8] V. Deineko, P. Jonsson, M. Klasson, and A. Krokhin. The approximability of Max CSP with fixed-value constraints. Technical Report CoRR cs.CC/0602075, 2006.

[9] A. Frank. Applications of submodular functions. In K. Walker, editor, *Surveys in Combinatorics*, volume 187 of *LMS Lecture Notes*, pages 85–136. Cambridge University Press, 1993.

[10] A. Frank. Matroids and submodular functions. In M. Dell'Amico, F. Maffoli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 65–80. John Wiley, 1997.

[11] S. Fujishige. *Submodular Functions and Optimization*, volume 58 of *Annals of Discrete Mathematics*. Elsevier, 2nd edition, 2005.

[12] A. Goldberg and R.E. Tarjan. A new approach to the maximum flow problem. *J. ACM*, 35:921–940, 1988.

[13] G. Grätzer. *General Lattice Theory*. Springer-Verlag, 2nd edition, 2002.

[14] G. Grätzer and D. Kelly. Products of lattice varieties. *Algebra Universalis*, 21:33–45, 1985.

[15] D. Hobby and R.N. McKenzie. *The Structure of Finite Algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, Providence, R.I., 1988.

[16] S. Iwata. A fully combinatorial algorithm for minimizing submodular functions. *Journal of Combinatorial Theory, Ser.B*, 84:203–212, 2002.

[17] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM*, 48(4):761–777, 2001.

[18] P. Jipsen and H. Rose. *Varieties of Lattices*, volume 1533 of *Lecture Notes in Mathematics*. Springer-Verlag, 1992. available from `www1.chapman.edu/~jipsen/JipsenRoseVoL.html`.

[19] P. Jonsson, M. Klasson, and A. Krokhin. The approximability of three-valued Max CSP. *SIAM Journal on Computing*, 35(6):1329–1349, 2006.

[20] A. Krokhin and B. Larose. Maximum constraint satisfaction on diamonds. Technical Report CS-RR-408, Department of Computer Science, University of Warwick, UK, 2004.

[21] A. Krokhin and B. Larose. Supermodular sublattices in products of relatively complemented lattices. manuscript, 2006.

[22] V.B. Lender. The groupoid of prevarieties of lattices. *Siberian Math. J.*, 16:1214–1223, 1975. (in Russian).

[23] A. Müller and D. Stoyan. *Comparison Methods for Stochastic Models and Risks.* Wiley & Sons, 2002.

[24] H. Narayanan. *Submodular Functions and Electrical Networks.* North-Holland, 1997.

[25] S.D. Promislow and V.R. Young. Supermodular functions on finite lattices. *Order*, 22:389–413, 2005.

[26] A. Schrijver. A combinatorial algorithm minimizing submodular functions in polynomial time. *Journal of Combinatorial Theory, Ser.B*, 80:346–355, 2000.

[27] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume B. Springer, 2003.

[28] D. Topkis. *Supermodularity and Complementarity.* Princeton University Press, 1998.