Finding Diverse Solutions in Combinatorial Problems with a Distributive Lattice Structure

Mark de Berg

Eindhoven University of Technology, The Netherlands

Andrés López Martínez

Eindhoven University of Technology, The Netherlands

Frits Spieksma

Eindhoven University of Technology, The Netherlands

Abstract

We generalize the polynomial-time solvability of k-Diverse Minimum s-t Cuts (De Berg et al., ISAAC'23) to a wider class of combinatorial problems whose solution sets have a distributive lattice structure. We identify three structural conditions that, when met by a problem, ensure that a k-sized multiset of maximally-diverse solutions – measured by the sum of pairwise Hamming distances – can be found in polynomial time. We apply this framework to obtain polynomial-time algorithms for finding diverse minimum s-t cuts, diverse stable matchings, and diverse market-clearing price vectors. Moreover, we show that the framework extends to two other natural measures of diversity. Lastly, we present a simpler algorithmic framework for finding a largest set of pairwise disjoint solutions in problems that meet these structural conditions.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Diversity, Lattice Theory, Submodular Function Minimization

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.11

Related Version *Full Version*: `https://arxiv.org/abs/2504.02369` [8]

Funding This research was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 945045, and by the NWO Gravitation project NETWORKS under grant no. 024.002.003.

1 Introduction

In combinatorial optimization problems, the objective is typically to identify a single optimal solution. However, this approach may be inadequate or impractical in real-world situations, where some constraints and factors are often overlooked or unknown in advance. This motivates the development of algorithms capable of finding multiple solutions, with *diversity* playing a key role. A growing body of research has focused on finding diverse solutions in classical combinatorial problems, much of it emerging from the field of fixed-parameter tractability [1, 9, 12, 23, 24, 32, 38, 29]. These studies show that finding diverse solutions is, in general, computationally more challenging than finding a single one. For instance, while Maximum Matching is solvable in polynomial time, finding two edge-disjoint perfect (or maximum) matchings is NP-hard, even on 3-regular graphs [12].

In this paper, we aim to develop theoretically efficient algorithms that produce a collection of maximally diverse solutions. We use the sum of pairwise Hamming distances between solutions as our measure of diversity. In contrast with the aforementioned literature, we show that a broader class of diverse problems is computationally no harder than finding a single solution in polynomial time. Specifically, we generalize the polynomial-time solvability of k-Diverse Minimum s-t Cuts by De Berg et al. [7] to a class of combinatorial problems whose solution sets form a distributive lattice.

We state our main result in terms of a unified general problem: Max-Sum k-Diverse Solutions. Let E be a finite set with n elements, and let $\Gamma \subseteq 2^E$ be a set of feasible solutions. For two feasible solutions $X, Y \in \Gamma$, the symmetric difference, or Hamming distance, between them is defined as $X \triangle Y = (X \setminus Y) \cup (Y \setminus X)$. Let $(X_1, X_2, \ldots, X_k)$ be a collection of k subsets of E. We consider the *pairwise-sum* diversity measure: $d_{\text{sum}}(X_1, X_2, \ldots, X_k) = \sum_{1 \leq i < j \leq k} |X_i \triangle X_j|$. We define Max-Sum k-Diverse Solutions as follows.

Max-Sum k-Diverse Solutions. *Given a finite set E of size n, an implicitly defined family Γ of subsets of E, referred to as feasible solutions, and a membership oracle for $\Gamma \subseteq 2^E$, find a k-multiset $C = (X_1, X_2, \ldots, X_k)$ with $X_1, X_2, \ldots, X_k \in \Gamma$, such that $d_{\text{sum}}(C)$ is maximum.*

Here, k is a fixed constant; i.e., k is not part of the input. Our main result is as follows.

▶ **Theorem 1.** *Max-Sum k-Diverse Solutions can be solved in polynomial time if the set of feasible solutions Γ satisfies the following three properties:*
1. *There is a relation $\leq$ such that the poset $(E, \leq)$ can be expressed as a disjoint union of r chains, and each feasible solution $X \in \Gamma$ contains exactly one element from each chain.*
2. *The set of feasible solutions with componentwise order defines a distributive lattice.*
3. *A compact representation of this lattice can be constructed in polynomial time.*

Similar to the approach of De Berg et al. [7], we achieve this result via a reduction to the *submodular function minimization* problem (SFM) on a distributive lattice, which is known to be solvable in polynomial time [20, 27, 36]. More precisely, we show that the pairwise-sum measure (reformulated as a minimization objective) is a submodular function on a distributive lattice of appropriately ordered k-sized collections of feasible solutions. By applying this result, in Section 4, we obtain polynomial-time algorithms for finding maximally diverse k-sized collections of stable matchings and market-clearing price vectors, while also reproducing the findings of De Berg et al. for minimum s-t cuts.

For simplicity, we present our results in terms of the d_{sum} measure. However, in Section 5 we will show that the framework extends to at least two other measures of diversity: the *coverage* (d_{cov}) and *absolute-difference* (d_{abs}) measures. Lastly, we consider the problem of finding a largest set of pairwise disjoint solutions in problems whose feasible solution set satisfies properties 1 and 2 of Theorem 1. In Section 6 we present an algorithm for this problem that bypasses the need for SFM.

Remark. Recently and independently from us, Iwamasa *et al.* [26] also presented a general framework for solving Max-Sum Diverse Solutions (and Max-Cov Diverse Solutions; see definition in Section 5) under certain conditions on the set of feasible solutions. They show that the conditions in our Theorem 1 imply the conditions in their framework. In the full version of this paper, we show that the reverse is true as well. Thus, their conditions and our conditions are, in fact, equivalent. Their approach reduces the problem to network flow, avoiding SFM and resulting in faster algorithms for computing diverse minimum s-t cuts and diverse stable matchings. However, in contrast to their framework, ours also supports the absolute-difference measure d_{abs} and thus, potentially applies to a broader range of problems.

2 Preliminaries

In this section, we introduce the notation and some basic results used throughout the paper. For a more comprehensive discussion on sets and posets, we refer to [25, 40], and for a detailed introduction to lattice theory, we refer to [2, 6, 19].

Sets, Multisets, and Tuples. For $k \in \mathbb{N}$, we use $[k]$ to denote the set $\{1, \dots, k\}$. The power set of a set M is denoted by 2^M. For any set M, we use the symbol M^k for the cartesian product; $\{(a_1, a_2, \dots, a_k) \mid a_i \in M\}$. The *disjoint union* of two sets is simply their union, but with the additional information that the two sets have no elements in common.

A *multiset* is a set in which elements can appear multiple times. The number of times an element appears in a multiset is referred to as its *multiplicity*. The *sum* of two multisets A and B, denoted by $A \uplus B$, is a multiset in which each element appears with a multiplicity equal to the sum of its multiplicity in A and in B. We refer to a multiset of cardinality k as a k-multiset. For a set M, we denote by M_k a k-multiset with elements drawn from M.

Unlike a multiset, where elements are unordered, a *tuple* is a collection of possibly repeated elements that is ordered. A k-tuple is a tuple of k elements. We denote a tuple by listing its elements within parenthesis and separated by commas; e.g., (a, b, c, d). Note that the cartesian product of k sets is a k-tuple.

Posets. A *partially ordered set (poset)* $P = (X, \preceq_P)$ consists of a ground set X along with a binary relation $\preceq_P$ on X that satisfies reflexivity, antisymmetry, and transitivity. When the relation $\preceq_P$ is evident from the context, we often use the same notation for both the poset and its ground set. In case a poset is indexed by a subscript i, we use $\preceq_i$ to denote its order relation. The Hasse diagram $G(P)$ of P, is a directed graph where each element of X is represented as a node, and an edge exists from element a to element b if $a \preceq_P b$ and no intermediate element c satisfies $a \preceq_P c \preceq_P b$. Vertices are arranged so that edge directions are implicitly understood as pointing upward.

A poset $P^* = (X^*, \preceq_P^*)$ is called a *subposet* of another poset $P = (X, \preceq_P)$ if (i) $X^* \subseteq X$ and (ii) for any $x, y \in X^*$ if $x \preceq_P^* y$ then $x \preceq_P y$. If the other direction of (ii) also holds, then we call P^* the subposet of P induced by X^*, and write $P^* = P[X^*]$. Given two posets $P = (X, \preceq_P)$ and $Q = (Y, \preceq_Q)$, their *disjoint union* $P \sqcup Q$ is the disjoint union of X and Y together with relation $\preceq_{P+Q}$ where $x \preceq_{P+Q} y$ if one of the following conditions holds: (i) $x, y \in X$ and $x \preceq_P y$, or (ii) $x, y \in Q$ and $x \preceq_Q y$. Thus, the Hasse diagram of $P \sqcup Q$ consists of the disconnected Hasse diagrams of P and Q drawn together.

A *chain* is a subset of a poset in which every pair of elements is comparable, and an *antichain* is a subset of a poset in which no two (distinct) elements are comparable. For any two elements x and y in a chain E with order relation $\preceq_E$, we say that x (resp. y) is a *chain-predecessor* (*chain-successor*) of y if $x \preceq_E y$. A poset is called a *chain decomposition* if the poset can be expressed as the disjoint union of chains.

For a poset $P = (X, \preceq_P)$, an *ideal* is a set $U \subseteq X$ where $u \in U$ implies that $v \in U$ for all $v \preceq_P u$. In terms of its Hasse diagram $G(P) = (X, E)$, a subset U of X is an ideal if and only if there is no incoming edge from U. We use $\mathcal{D}(P)$ to denote the family of all ideals of P. If $x \preceq_P y$ in the poset, then the closed interval from x to y, denoted by $[x, y]$, is the poset with ground set $\{z \in X \mid x \preceq_P z \preceq_P y\}$ together with relation $\preceq_P$.

Now we introduce the notion of *componentwise order*. Let $(X_i, \preceq_i)$, $i \in [r]$ be posets, with r a positive integer, and let $Y \subseteq X_1 \times \dots \times X_r$. The componentwise order on Y is an order relation $\preceq$ defined as follows: Given two tuples $(a_1, a_2, \dots, a_r)$ and $(b_1, b_2, \dots, b_r) \in Y$, we write $(a_1, a_2, \dots, a_r) \preceq (b_1, b_2, \dots, b_r)$ iff $a_i \preceq_i b_i$ for all $i \in [r]$. Note that we drop the subscript in $\preceq$ whenever the order relation is a component-wise order. If the posets $(X_i, \preceq_i)$, $i \in [r]$, are all the same poset $(X, \preceq)$, we use $\preceq^r$ to denote the componentwise order on X^r and refer to it as the *product order*.

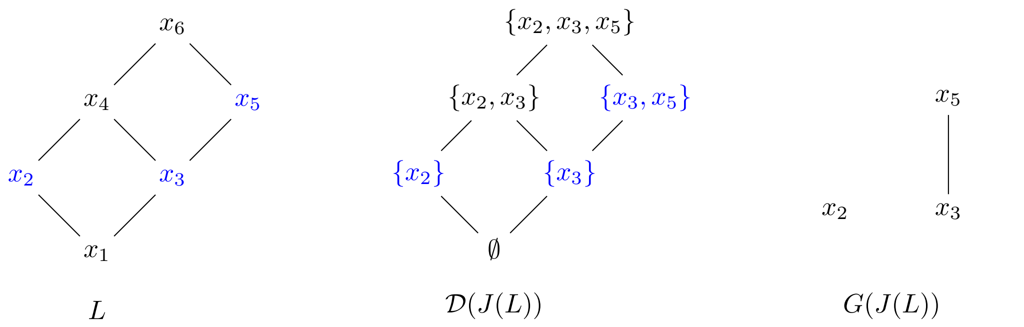

Figure 1 Example of Birkhoff's representation theorem for distributive lattices. The left is a distributive lattice L, the middle is the isomorphic lattice $\mathcal{D}(J(L))$ of ideals of join-irreducibles of L, and the right shows the compact representation $G(J(L))$ of L. The join irreducible elements of L and $\mathcal{D}(J(L))$ are highlighted in blue.

Lattices. A *lattice* is a poset $L = (X, \preceq)$ in which any two elements $x, y \in X$ have a (unique) greatest lower bound, or *meet*, denoted by $x \wedge y$, as well as a (unique) least upper bound, or *join*, denoted by $x \vee y$. We can uniquely identify L by the tuple $(X, \vee, \wedge)$. The *bottom*, or minimum, element in the lattice L is denoted by $0_L := \bigwedge_{x \in L} x$. Likewise, the *top*, or maximum, element of L is given by $1_L := \bigvee_{x \in L} x$. A lattice L' is a *sublattice* of L if $L' \subseteq L$ and L' has the same meet and join operations as L. In this paper we only consider *distributive lattices*, which are lattices whose meet and join operations satisfy distributivity; that is, $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ and $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$, for any $x, y, z \in L$. Note that a sublattice of a distributive lattice is also distributive. Every chain is a distributive lattice with max as join ($\vee$) and min as meet ($\wedge$).

Suppose we have a collection $L_1, \dots, L_k$ of lattices $L_i = (X_i, \vee_i, \wedge_i)$ with $i \in [k]$. The *(direct) product lattice* $L_1 \times \ldots \times L_k$ is a lattice with ground set $X = \{(x_1, \dots, x_k) : x_i \in L_i\}$ and join $\vee$ and meet $\wedge$ operations acting component-wise; that is, $x \vee y = (x_1 \vee_1 y_1, \dots, x_k \vee_k y_k)$ and $x \wedge y = (x_1 \wedge_1 y_1, \dots, x_k \wedge_k y_k)$ for any $x, y \in X$. The lattice L^k is the product lattice of k copies of L, and is called the k-th *power* of L. If L is a distributive lattice, then L^k is also distributive.

A crucial notion in this paper is that of *join-irreducibles*. An element x of a lattice L is called *join-irreducible* iff $x \neq 0_L$ and it cannot be expressed as the join of two elements $y, z \in L$ with $y, z \neq x$. In a lattice, any element is equal to the join of all join-irreducible elements lower than or equal to it. The set of join-irreducible elements of L is denoted by $J(L)$. Note that $J(L)$ is a poset whose order is inherited from L. Due to Birkhoff's representation theorem – a fundamental tool for studying distributive lattices – every distributive lattice L is isomorphic to the lattice $\mathcal{D}(J(L))$ of ideals of its poset of join-irreducibles, with union and intersection as join and meet operations. See Figure 1 for an illustration.

▶ **Theorem 2** (Birkhoff's Representation Theorem [2])**.** *Any distributive lattice L can be represented as the poset of its join-irreducibles $J(L)$, with the order induced from L.*

Hence, a distributive lattice L can be uniquely recovered from $J(L)$ and can thus be represented as the Hasse diagram of its poset of join-irreducibles, denoted by $G(J(L))$. We refer to $G(J(L))$ as a *compact representation* of L, since $J(L)$ is usually exponentially smaller than L. This representation is useful when designing algorithms, as the size of $G(J(L))$ is $O(|J(L)|^2)$, while L can have as many as $2^{|J(L)|}$ elements. Keep in mind, however, that Theorem 2 only guarantees the existence of such a compact representation; it does not provide a method to efficiently find the set $J(L)$.

Submodular Function Minimization. Let $f : X \to \mathbb{R}$ be a real-valued function on a lattice $L = (X, \preceq)$. We say that f is *submodular* on L if $f(x \wedge y) + f(x \vee y) \leq f(x) + f(y)$, for all $x, y \in X$. If $-f$ is submodular in L, then we say that f is *supermodular* in L and just *modular* if f is both sub and supermodular. The *submodular function minimization* problem (SFM) on lattices is, given a submodular function f on L, to find an element $x \in L$ such that $f(x)$ is minimum. An important fact that we use in our work is that the sum of submodular functions is also submodular. Also, note that minimizing f is equivalent to maximizing $-f$. It is known that a submodular function on a distributive lattice L can be minimized in polynomial time in $|J(L)|$ – the number of join-irreducibles of L [20, 27, 36, 4].

▶ **Theorem 3** ([34, Note 10.15] and [30, Thm.1])**.** *For any distributive lattice L, given by its poset of join-irreducibles $J(L)$, a submodular function $f : L \to \mathbb{R}$ can be minimized in polynomial time in $|J(L)|$, provided a polynomial time evaluation oracle for f.*

3 The Reduction to SFM

In this section, we prove Theorem 1 by reducing Max-Sum k-Diverse Solutions to SFM on a distributive lattice, under the assumption that the feasible solution set satisfies properties 1–3 of the theorem. The core ideas of this proof were already established by De Berg et al. [7] in the context of minimum s-t cuts. We include the argument here for completeness, while adapting and generalizing it to a broader setting.

The proof is divided into four parts, each supported by a corresponding lemma. The Distributivity Lemma (Lemma 4) shows that the set of (yet to be defined) *left-right ordered* k-tuples of feasible solutions, with product order, defines a distributive lattice L^*. The Cost-Equivalence Lemma (Lemma 7) further shows that optimizing the diversity over this lattice is the same as optimizing over the original set Γ_k of k-multisets of Γ. Hence, we can restrict ourselves to the elements of L^*. Next, the Submodularity Lemma (Lemma 11) establishes that the pairwise-sum measure (reformulated as a minimization objective) is a submodular function on L^*. Finally, the Compactness Lemma (Lemma 12) ensures that a compact representation of L^* can be constructed in polynomial time.

We begin by establishing some consequences of properties 1–3 of Theorem 1. Consider a ground set E and a set of feasible solutions $\Gamma \subseteq 2^E$ for which the properties hold. By property 1, we know that there is a poset associated to E that is the disjoint union of r chains $(E_i, \preceq_i)$, $i \in [r]$, and that each feasible solution $X \in \Gamma$ contains exactly one element from each chain, meaning $|X| = r$ and $\Gamma \subseteq E_1 \times \cdots \times E_r$. Then, the set Γ, with componentwise order $\preceq$, forms a poset of feasible solutions $L = (\Gamma, \preceq)$. Furthermore, by properties 1 and 2, this poset is a distributive lattice, with join ($\vee$) and meet ($\wedge$) given by componentwise maximum and minimum. Let us now consider the poset $(\Gamma^k, \preceq^k)$ of k-tuples of feasible solutions, with product order $\preceq^k$. We say that a k-tuple $C = (X_1, X_2, \ldots, X_k)$ in Γ^k is in *left-right order* if $X_i \preceq X_j$ for all $i < j$. That is, the feasible solutions in C are arranged in non-decreasing order according to relation $\preceq$. Let $\Gamma^k_{\text{lr}} \subseteq \Gamma^k$ denote the subset of left-right ordered k-tuples.

Part 1: Distributivity. We now establish the first of the four lemmas.

▶ **Lemma 4** (Distributivity Lemma)**.** *The poset $L^* = (\Gamma^k_{\text{lr}}, \preceq^k)$ is a distributive lattice.*

Proof. By property 2 of Theorem 1, $L = (\Gamma, \preceq)$ is a distributive lattice. Now, let $L^k = (\Gamma^k, \preceq^k)$ be the k-th power of L. We know that the product of distributive lattices is distributive; hence L^k is a distributive lattice. Moreover, since $\Gamma^k_{\text{lr}} \subseteq \Gamma^k$, the poset L^* is a sublattice of L^k. As any sublattice of a distributive lattice is itself distributive, the lemma follows. ◀

Part 2: Cost equivalence. Following the proof of the Distributivity Lemma, we now establish an equivalence between the costs of maximum diversity solutions in the sets Γ^k and Γ^k_{lr}. (Note that this is the same as establishing the equivalence between the sets Γ_k and Γ^k_{lr}, since a k-multiset over Γ has the same diversity as each of its up to $n!$ permutations – each a k-tuple – in Γ^k.) For this, we use the notion of element multiplicity. Let $C \in \Gamma^k$ be a k-tuple of solutions. The multiplicity $\mu_e(C)$ of an element $e \in E$, with respect to C, is the number of feasible solutions in C that contain e. Since a feasible solution contains no repeated elements, $\mu_e(C)$ is also the number of times e appears in the multiset sum of the solutions in C. An immediate consequence of property 1 of Theorem 1 is the following.

▶ **Observation 5.** *For any two $X, Y \in L$, we have $X \uplus Y = (X \wedge Y) \uplus (X \vee Y)$.*

Proof. Let $X = (x_1, \dots, x_r)$ and $Y = (y_1, \dots, y_r)$, with $x_i, y_i \in E_i$ and $i \in [r]$, where E_i is the i-th chain in the chain decomposition of E. By property 1, we know that the meet and join of two elements L are given by componentwise minimum and maximum. That is,

$$X \wedge Y = (\min(x_1, y_1), \dots, \min(x_r, y_r)), \quad \text{and} \quad X \vee Y = (\max(x_1, y_1), \dots, \max(x_r, y_r)).$$

Hence, if $x_i = y_i$, the element x_i appears twice in the multiset sum $X \uplus Y$ and twice in the sum $(X \wedge Y) \uplus (X \vee Y)$. If $x_i \neq y_i$, then x_i appears in either the join or the meet of X and Y, and similarly for element y_i. Finally, if an element $e \in E$ is not in $X \cup Y$, then it is neither the minimum nor maximum of any entry and therefore cannot appear in $(X \wedge Y) \cup (X \vee Y)$. Since this holds for each $i \in [k]$, the observation is proven. ◀

Observation 5 implies that the join and meet operations of the lattice L of feasible solutions preserve element multiplicities. Consequently, any k-tuple in Γ^k can be reordered into a left-right ordered form while preserving element multiplicities, as stated in the following claim (see the proof in the full version).

▷ Claim 6. For every k-tuple $C \in \Gamma^k$ there exists a left-right ordered k-tuple $\hat{C} \in L^k_{\mathrm{lr}}$ such that $\mu_e(C) = \mu_e(\hat{C})$ for all $e \in E$.

Now, consider the pairwise-sum diversity measure introduced in Section 1. We can rewrite it directly in terms of the multiplicity as

$$d_{\mathrm{sum}}(C) = \sum_{e \in E} \mu_e(C)(k - \mu_e(C)) = k^2 \cdot r - \sum_{e \in E} \mu_e(C)^2, \tag{1}$$

which results from observing that $\sum_{e \in E} \mu_e(C) = k \cdot r$, for any $C \in \Gamma^k$, with r the cardinality of each solution in C. This formulation highlights that maximizing d_{sum} depends only on the distribution of elements across feasible solutions, rather than their specific ordering within a tuple C. Since the terms outside the latter summation in Eq. (1) are constant, the following lemma is an immediate consequence of Claim 6.

▶ **Lemma 7** (Cost-Equivalence Lemma). *For any $C \in \Gamma^k$ there exists $\hat{C} \in \Gamma^k_{\mathrm{lr}}$ such that $d_{\mathrm{sum}}(\hat{C}) = d_{\mathrm{sum}}(C)$.*

Lemma 7 tells us that in order to solve Max-Sum k-Diverse Solutions, we do not need to optimize over the set of k-element multisets of Γ. Instead, we can optimize over the set Γ^k_{lr} of k-tuples that are in left-right order. Moreover, it follows from Eq. (1) that maximizing d_{sum} is equivalent to minimizing

$$\hat{d}_{\mathrm{sum}}(C) = \sum_{e \in E} \mu_e(C)^2. \tag{2}$$

Hence, solving Max-Sum k-Diverse Solutions reduces to minimizing $\hat{d}_{\text{sum}}(C)$ in the lattice L^*. All we have left to do to complete the reduction to SFM is show that $\hat{d}_{\text{sum}}(C)$ is submodular in the lattice L^*.

Part 3: Submodularity. We begin with two claims regarding the multiplicity function $\mu_e(C)$ on L^*. These claims rely crucially on property 1 of Theorem 1. Their proofs can be found in the full version of the paper. We use $E(C)$ to denote the set of elements $\bigcup_{X \in C} X$ for a tuple $C \in \Gamma^k$.

▷ Claim 8. The multiplicity function $\mu_e : \Gamma^k_{\text{lr}} \to \mathbb{N}$ is modular on L^*.

▷ Claim 9. For any two $C_1, C_2 \in L^*$ and $e \in E(C_1) \cup E(C_2)$, it holds that $\max(\mu_e(C_1 \vee C_2), \mu_e(C_1 \wedge C_2)) \leq \max(\mu_e(C_1), \mu_e(C_2))$.

We observe that Claim 8 holds in the lattice L^k, not just in the sublattice L^* of left-right ordered k-tuples. In contrast, Claim 9 is specific to the sublattice L^*. The following proposition is an immediate consequence of these two claims and the convexity of the square function.

▶ **Proposition 10.** *For any two $C_1, C_2 \in L^*$ and any $e \in E$, we have $\mu_e(C_1 \vee C_2)^2 + \mu_e(C_1 \wedge C_2)^2 \leq \mu_e(C_1)^2 + \mu_e(C_2)^2$.*

Proposition 10 states that, for each element $e \in E$, the square of the multiplicity function μ_e is submodular in the lattice L^*. Then, taking the sum of $\mu_e(C)^2$ over all elements $e \in E$ is also a submodular function; that is

$$\sum_{e \in E} \mu_e(C_1 \vee C_2)^2 + \sum_{e \in E} \mu_e(C_1 \wedge C_2)^2 \leq \sum_{e \in E} \mu_e(C_1)^2 + \sum_{e \in E} \mu_e(C_2)^2.$$

Each sum in this inequality corresponds to the definition of $\hat{d}_{sum}$ applied to the arguments $C_1 \vee C_2$, $C_1 \wedge C_2$, C_1 and C_2, respectively. Hence, we obtain the following.

▶ **Lemma 11** (Submodularity Lemma). *The function $\hat{d}_{\text{sum}} : \Gamma^k_{\text{lr}} \to \mathbb{N}$ is submodular in L^*.*

Part 4: Compactness. While Lemmas 4, 7, and 11 already demonstrate the reduction of Max-Sum k-Diverse Solutions to SFM, this reduction alone does not guarantee an efficient algorithm. To complete the proof of Theorem 1, it remains to show that a compact representation of the left-right ordered lattice L^* exists and can be constructed efficiently. By Birkhoff's representation theorem, we need only specify the set of join-irreducibles of L^* to obtain a compact representation in $O(|J(L^*)|^2)$ time. This is done in the following lemma, whose proof can be found in the full version of this work.

▶ **Lemma 12** (Compactness Lemma). *The set of join-irreducibles of L^* is of size $O(kn)$ and is given by*

$$J(L^*) = \textstyle\bigcup_{i=1}^k J_i, \text{ where } J_i := \{(\underbrace{0_L, \ldots, 0_L}_{i-1 \text{ times}}, \underbrace{p, \ldots, p}_{k-i+1 \text{ times}}) : p \in J(L)\}.$$

With Lemma 12, a compact representation of L^* can be constructed in polynomial time. It is also clear that, given a k-tuple, the function $\hat{d}_{\text{sum}}$ can be evaluated efficiently. Then, by Theorem 3 and Lemmas 4, 7, 11, and 12, the proof of Theorem 1 is complete.

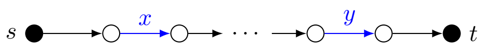

Figure 2 Illustration of the order relation $\preceq_i$ over the edges of an s-t path $p_i \in \mathcal{P}$.

4 Applications of the Framework

We now present examples of combinatorial problems whose feasible solution sets meet each of the conditions outlined in Theorem 1, allowing for the generation of maximally diverse solutions within our framework. Specifically, we discuss minimum s-t cuts (Section 4.1), stable matchings (Section 4.2), and market-clearing price vectors (Section 4.3).

4.1 Minimum s-t cuts

In the Minimum s-t Cut problem we are given a directed graph $G = (V, E)$, along with two special vertices $s, t \in V$, and are tasked with finding a subset $S \subseteq E$ of minimum cardinality that separates vertices s and t, meaning that removing these edges from G ensures there is no path from s to t. Such a set is called a *minimum s-t cut* or *s-t mincut*. Here, we consider the problem of finding diverse minimum s-t cuts, formally defined below.

Max-Sum k-Diverse Minimum s-t Cuts. *Given are a directed graph $G = (V, E)$ and vertices $s, t \in V$. Let $\Gamma \subseteq 2^E$ be the set of minimum s-t cuts in G, and let Γ_k be the set of k-multisets over Γ. Find $C \in \Gamma_k$ such that $d_{\text{sum}}(C) = \max_{S \in \Gamma_k} d_{\text{sum}}(S)$.*

Using our framework, we reproduce the findings of De Berg et al. [7] for Max-Sum k-Diverse Minimum s-t Cuts by showing that the set Γ of minimum s-t cuts satisfies properties 1-3 of Theorem 1. We prove these statements in order.

▶ **Lemma 13** (Property 1). *There is a chain decomposition of the edge set E into r disjoint chains, such that each minimum s-t cut $X \in \Gamma$ contains exactly one element from each chain.*

Proof. We construct the r chains as follows. Let $\mathcal{P}$ be an (arbitrary) set of edge-disjoint s-t paths in G with maximum cardinality r. Define $E(p_i)$ as the set of edges traversed by the path $p_i \in \mathcal{P}$. For each path $p_i \in \mathcal{P}$, consider the order relation $\preceq_i$ defined as follows: for any $x, y \in E(p_i)$, we say $x \preceq_i y$ if and only if path p_i meets edge x before edge y, or if x and y are the same edge. Since every pair of edges within a path p_i is comparable under this relation, each poset $(E(p_i), \preceq_i)$, for $i \in [r]$, forms a chain. Moreover, these chains are disjoint by the definition of the set $\mathcal{P}$. By Menger's theorem, the size of a minimum s-t cut in G equals the maximum number of edge-disjoint s-t paths, which is r. Consequently, any minimum s-t cut $X \subseteq E$ must include exactly one edge from each chain $(E(p_i), \preceq_i)$, $i \in [r]$. Otherwise, if X contained fewer than r edges, it would not be a valid s-t cut, and if it contained more, it would not be of minimum size. Hence, $\Gamma \subseteq E(p_1) \times \cdots \times E(p_r)$.

Consider now the edges in $E' = E \setminus \bigcup_{1 \leq i \leq r} E(p_i)$. We call these edges *residual edges*. Observe that these edges can never be part of a minimum s-t cut. This follows because such a cut must contain exactly one edge from each chain in $\mathcal{P}$, and cutting any additional edge from E' would only increase the cut size, violating minimality. Hence, we simply distribute the residual edges arbitrarily over the r chains. This does not change the fact that the chains are disjoint, or that the set of minimum s-t cuts is a subset of the cartesian product of the augmented chains. This completes the proof. ◀

By Lemma 13, the set $\Gamma \subseteq E(p_1) \times \cdots \times E(p_r)$ of minimum s-t cuts with componentwise order – defined by: $(x_1, \ldots, x_r) \preceq (y_1, \ldots, y_r)$ for $(x_1, \ldots, x_r), (y_1, \ldots, y_r) \in \Gamma$ iff $x_i \preceq_i y_i$ for all $i \in [r]$ – forms a poset $L = (\Gamma, \preceq)$. It is well known that this poset defines a distributive lattice [11, 31, 22]. Specifically, proving that Γ is closed under the joins and meets induced by $\preceq$ suffices to establish this property (see e.g., [7, Claim A.1]). Thus, property 2 of Theorem 1 follows directly.

▶ **Lemma 14** (Property 2)**.** *The set Γ of minimum s-t cuts with componentwise order $\preceq$ defines a distributive lattice L.*

Next, we note that a compact representation of the lattice of minimum s-t cuts can be constructed in polynomial time. This result is well known from the work of Picard and Queyranne [35], who gave an algorithm to build such a representation using a residual graph. Specifically, the resulting graph has vertex set $J(L) \cup 0_L$ and total size at most $|V|$, and can therefore be constructed in $O(|V|^2)$ time.

▶ **Lemma 15** (Property 3)**.** *A compact representation of the lattice s-t mincuts can be constructed in polynomial time.*

Then, by Theorem 1 and Lemmas 13-15, we obtain a polynomial time algorithm for Max-Sum k-Diverse Minimum s-t Cuts via submodular function minimization.

▶ **Theorem 16.** *Max-Sum k-Diverse Minimum s-t Cuts is poly-time solvable.*

▶ Remark 17. Since the edge version of Menger's theorem is known to hold for multigraphs, our results for unweighted directed graphs extend naturally to weighted graphs by replacing each edge of weight w with w parallel edges.

▶ Remark 18. Similar results to those presented in Lemmas 13-15 can be established for *minimum s-t vertex cuts*. Since a vertex-connectivity version of Menger's theorem also exists, the arguments in Lemma 13 remain valid when replacing E with V. Moreover, the poset of minimum s-t vertex cuts, ordered componentwise, forms a distributive lattice, which can be demonstrated analogously to Lemma 14. Finally, a compact representation of this lattice can be computed efficiently, as shown by Bonsma [5, Sec. 6], or via the constructive version of Birkhoff's theorem for computing a *slice*, as described by Garg [33] (see also [15, Ch. 10]).

4.2 Stable Matchings

In the Stable Matching problem, we are given a complete bipartite graph $K_{n,n} = (A \cup B, E)$ along with a linear ordering $\preceq_a$ over B for each vertex $a \in A$, and similarly a linear ordering $\preceq_b$ over A for each vertex $b \in B$. For a vertex $a \in A$ (resp. $b \in B$), the poset $L_a = (B, \preceq_v)$ (resp. $L_b = (A, \preceq_v)$) is referred to as its *preference list*. The task is to find a perfect matching M in $K_{n,n}$ such that no two vertices $a \in A$ and $b \in B$ prefer each other over their matched partners. Such a set of edges is called a *stable matching*. We now consider the problem of finding diverse stable matchings.

Max-Sum k-Diverse Stable Matching. *Given are a complete bipartite graph $K_{n,n} = (A \cup B, E)$, along with preference lists L_a and L_b for each $a \in A$ and $b \in B$. Let $\Gamma \subseteq 2^E$ be the set of stable matchings in G, and let Γ_k denote the set of k-multisets over Γ. Find $C \in \Gamma_k$ such that $d_{\text{sum}}(C) = \max_{S \in \Gamma_k} d_{\text{sum}}(S)$.*

We show that Max-Sum k-Diverse Stable Matching can be solved in polynomial time by proving that the set Γ of stable matchings satisfies properties 1-3 of Theorem 1.

▶ **Lemma 19** (Property 1)**.** *There is a chain decomposition of the edge set E into r disjoint chains, such that each stable matching $X \in \Gamma$ contains exactly one element from each chain.*

Proof. Let $r = n$. Note that the posets L_a and L_b are chains. We claim that the chains L_a, $a \in A$ define a disjoint chain decomposition of the ground set E.[1] First, we argue for disjointness. Let $E(a) = \{(a, b) \mid b \in L_a\}$ denote the set of edges defined by the preference list L_a of an arbitrary vertex $a \in A$. Since $K_{n,n}$ is bipartite, there are no edges between the vertices of A. This implies that $E(a_1) \cap E(a_2) = \emptyset$ for all distinct $a_1, a_2 \in A$. Moreover, $E = \bigcup_{a \in A} E(a)$. Hence, the chains L_a, $a \in A$ define a disjoint chain decomposition of E.

Now, we argue that a stable matching must contain exactly one element from each chain L_a. This follows immediately from the definition of perfect matching, which requires every vertex in A to be matched to exactly one vertex in B. Consequently, each stable matching selects precisely one edge from $E(a)$ for each $a \in A$. This completes the proof. ◀

To establish property 2, we use the following well-known result from the stable matchings literature [28, 3].

▷ Claim 20 ([28, Thm. 7 & Cor. 1]). Given any two stable matchings $X = ((a_1, b_1), \ldots, (a_n, b_n))$ and $Y = ((a_1, b'_1), \ldots, (a_n, b'_n))$, then

$$X \vee Y = ((a_1, \max_{\preceq_{a_1}}(b_1, b'_1)), \ldots, (a_n, \max_{\preceq_{a_n}}(b_n, b'_n)))) \quad \text{and}$$
$$X \wedge Y = ((a_1, \min_{\preceq_{a_1}}(b_1, b'_1)), \ldots, (a_n, \min_{\preceq_{a_n}}(b_n, b'_n))))$$

are also stable matchings.

By standard results in lattice theory (see e.g., [19]), the cartesian product $E_{\text{prod}} = E(a_1) \times \cdots \times E(a_n)$, with componentwise order $\preceq$, forms a distributive lattice $(E_{\text{prod}}, \preceq)$. Then, by Lemma 19 and Claim 20, the poset $L = (\Gamma, \preceq)$ is a sublattice of E_{prod}, which implies that L is also distributive.

▶ **Lemma 21** (Property 2). *The set Γ of stable matchings with componentwise order $\preceq$ defines a distributive lattice L.*

It only remains to verify that property 3 of Theorem 1 is satisfied by the set Γ of stable matchings. This property follows directly, since the so-called *poset of rotations* introduced by Gusfield [21] provides the required structure (see also, e.g., [14, Sec 2.3]).

▶ **Lemma 22** (Property 3 [21, Lemma 3.3.2]). *A compact representation of the lattice L of stable matchings can be constructed in $O(|V|^2)$ time.*

Then, by Theorem 1 and Lemmas 19-22, the following theorem holds.

▶ **Theorem 23.** *Max-Sum k-Diverse Stable Matching is poly-time solvable.*

4.3 Market Clearing Price Vectors

As a final example, we consider the problem of finding an integer *market-clearing price* in a *matching market* (see e.g., [10, Ch. 10] for details). Such a market consists of a set I of n items and a set U of n bidders, where each item can be assigned to at most one bidder, and each bidder wants to buy at most one item. Each bidder $b \in U$ assigns a valuation $v_{b,i}$ to every item $i \in I$, where $v_{b,i}$ is an integer between 0 and T. We assume that $T = poly(n)$.

[1] Note that we may also choose the chains L_b, $b \in B$ and get similar results.

The Market Clearing Price problem asks for a price vector P assigning a price $P[i] \in [0, T]$ to each item i, such that the bipartite graph $(I, U, E(P))$ defined by

$$(j, b) \in E(P) \iff \forall i \in I : (v_{b,j} - P[j]) \geq (v_{b,i} - P[i])$$

has a perfect matching. (Informally, an edge connects item j to bidder b if j gives b their highest payoff – valuation minus price.) A price vector P that yields such a graph is called a *market-clearing price* vector.

We now turn to the problem of finding diverse market-clearing price vectors.

Max-Sum k-Diverse Market Clearing Price. *Given is a matching market $M = (I, U, v)$ where I is a set of n items, U is a set of n bidders, and $v : I \times U \to [0, T]$ is a valuation function. Let $\Gamma \subseteq [T]^n$ be the set of market-clearing price vectors in M, and let Γ_k be the set of k-multisets over Γ. Find $C \in \Gamma_k$ such that $d_{\text{sum}}(C) = \max_{S \in \Gamma_k} d_{\text{sum}}(S)$.*

At this point, the reader may wonder whether d_{sum} is a good choice for vectors; we return to this question later. For now, we show that Max-Sum k-Diverse Market Clearing Price can be solved in polynomial time by verifying that Γ satisfies properties 1–3 of Theorem 1. The first property follows immediately from the definition of a price vector.

▶ **Lemma 24** (Property 1). *There is a chain decomposition of the ground set E into r disjoint chains, such that each market-clearing price vector contains exactly one element from each chain.*

Proof. By definition, a market-clearing price vector is an element of the power set $E_{\text{prod}} = [0, T]^n$. Hence, the chain decomposition of E_{prod} consists of n copies of the integer interval $E = [0, T]$, with elements from different chains deemed incomparable. The ordering $\preceq_i$ of each chain E_i is the natural ordering. ◀

It is clear that the power set E_{prod} with product order $\preceq^n$ forms a distributive lattice, where join ($\vee$) and meet ($\wedge$) are determined by componentwise maximum and minimum. Shapley et al. [37] (see also [39]) establish that the poset $L = (\Gamma, \preceq^n)$ is closed under these join and meet operations, hence L is a sublattice of E_{prod}, and thus, L is distributive.

▶ **Lemma 25** (Property 2 [37]). *The poset $(\Gamma, \preceq^n)$ is a distributive lattice, with the meet and join defined appropriately.*

As for property 3, Garg [16, 17] recently established that the set $J(L)$ of join-irreducibles of the lattice L of market-clearing price vectors can be determined efficiently (in polynomial time) via an algorithm for detecting so-called *lattice-linear predicates* (see also [18] and [15, Ch. 10]). We thus get the following result.

▶ **Lemma 26** (Property 3 [17]). *A compact representation of the lattice L of market-clearing price vectors can be constructed in polynomial time.*

Then, by Theorem 1 and Lemmas 24, 25, and 26, the following theorem holds.

▶ **Theorem 27.** *Max-Sum k-Diverse Market-Clearing Price is poly-time solvable.*

Let us briefly reflect on the choice of diversity measure. The pairwise-sum measure d_{sum} captures whether elements differ, but not by how much; an important aspect for price vectors, where the magnitude of values matters more than their identity (unlike in cuts or matchings). In the next section, we extend our framework to support alternative measures, such as the *absolute-difference* measure, which better captures diversity in price vectors.

5 Other Diversity Measures

The proof of Theorem 1 relies on four lemmas, with the diversity measure playing a role in only two of them: the Cost-Equivalence (Lemma 7) and Submodularity (Lemma 11) lemmas. For simplicity, we have presented our main result in terms of the d_{sum} diversity measure. However, the framework is not limited to this specific measure. Just as it applies to problems whose solution sets satisfy the properties of Theorem 1, it also extends to other diversity measures, provided they satisfy both the Cost-Equivalence and Submodularity lemmas.

Here, we mention two such diversity measures: the *coverage* diversity d_{cov}, and the L_1- or *absolute-difference* diversity d_{abs}. Let E be a finite set with n elements, and let $\Gamma \subseteq 2^E$ be a set of feasible solutions. Given a k-tuple of feasible solutions $(X_1, \dots, X_k) \in \Gamma^k$, these measures are defined as follows:

$$d_{\text{cov}}(X_1, X_2, \dots, X_k) = \bigcup_{1 \leq i \leq k} |X_i|, \quad \text{and} \tag{3}$$

$$d_{\text{abs}}(X_1, X_2, \dots, X_k) = \sum_{1 \leq i < j \leq k} f(X_i, Y_j), \tag{4}$$

where $f(X, Y) = \sum_i^r \|x_i - y_i\|$ for any two $X = (x_1, \dots, x_r), Y = (y_1, \dots, y_r) \in \Gamma$.

The coverage diversity measures the number of distinct elements appearing across solutions, while the absolute-difference diversity quantifies diversity by summing coordinate-wise differences between solutions. Notice that the latter applies only to solutions representable as r-tuples, since f requires component-wise comparisons, and it assumes a notion of difference between elements in E (e.g., r-dimensional integer vectors in $[-M, M]^r$, with $M \in \mathbb{N}$).

Each of these two measures gives rise to a corresponding optimization problem, defined in the same way as Max-Sum Diverse Solutions but with d_{sum} replaced by d_{cov} or d_{abs}. We refer to these as Max-Cov k-Diverse Solutions and Max-Abs k-Diverse Solutions, respectively. Our main result for these problems is the following:

▶ **Theorem 28.** *Max-Cov k-Diverse Solutions and Max-Abs k-Diverse Solutions can be solved in polynomial time if the set of feasible solutions Γ satisfies the three properties of Theorem 1.*

We establish this result by showing that both d_{cov} and d_{abs} satisfy the Cost-Equivalence and Submodularity lemmas from Section 3. Due to space constraints, the proofs for d_{cov} are deferred to the full version (see also [7, Thm. 14]). Here, we prove these lemmas for d_{abs}.

Proof of Cost-Equivalence. The following lemma is an immediate consequence of Claim 6, which states that any k-tuple in Γ^k can be reordered into a left-right ordered form while preserving element multiplicities.

▶ **Lemma 29** (Cost-Equivalence Lemma). *Let $C \in \Gamma^k$ such that $d_{abs}(C) = \max_{S \in \Gamma^k} d_{\text{abs}}(S)$. Then there exists $\hat{C} \in \Gamma^k_{\text{lr}}$ such that $d_{\text{abs}}(\hat{C}) = d_{\text{abs}}(C)$.*

Proof. Let $C \in \Gamma^k$ be an arbitrary k-tuple of solutions, and let $\hat{C} \in \Gamma^k_{\text{lr}}$ be its reordering into left-right order by the algorithm of Claim 6. For a feasible solution $X \in \Gamma$, let $X(\ell)$ denote its ℓ-th component.

Consider the k-tuples $C(\ell) = (X_1(\ell), \dots, X_k(\ell))$ and $\hat{C}(\ell) = (\hat{X}_1(\ell), \dots, \hat{X}_k(\ell))$, where $X_i \in C$, $\hat{X}_i \in \hat{C}$ for all $i \in [k]$. These represent the ℓ-th component of each solution in C and $\hat{C}$, respectively. Now, define the function $f_\ell : E^k_\ell \to \mathbb{R}$ as:

$$f_\ell(x_1, \dots, x_k) = \sum_{1 \leq i < j \leq k} \|x_i - x_j\| .$$

By Claim 6, the multiplicity of each element in $C(\ell)$ is preserved in $\hat{C}(\ell)$, implying that $f_\ell(C(\ell)) = f_\ell(\hat{C}(\ell))$. Since the absolute-difference diversity measure decomposes as:

$$d_{\text{abs}}(X_1, \ldots, X_k) = \sum_{\ell=1}^{r} f_\ell(X_1(\ell), \ldots, X_k(\ell)),$$

it follows that $d_{\text{abs}}(\hat{C}) = d_{\text{abs}}(C)$. In particular, this holds for tuples that achieve maximum diversity. ◀

Proof of Submodularity. In this case, the Submodularity Lemma actually becomes a Modularity Lemma. First, consider the function $f'_\ell : \Gamma^2_{\text{lr}} \to \mathbb{R}$ defined by $f'_\ell(X_1, X_2) = \|X_1(\ell) - X_2(\ell)\|$, where $X_1 \preceq X_2$, $\ell \in [r]$, and $X(\ell)$ denotes the ℓ-th component of a solution $X \in \Gamma$. We can rewrite the absolute difference diversity measure as:

$$d_{\text{abs}}(X_1, \ldots, X_k) = \sum_{\ell}^{r} \sum_{1 \le i < j \le k} f'_\ell(X_i, X_j).$$

If we can establish that $f'_\ell(\cdot)$ is modular in L, then, because the sum of modular functions is modular, d_{abs} would also be modular. We prove this in the following lemma.

▶ **Lemma 30** (Modularity Lemma). *The function $\hat{d}_{\text{abs}} : \Gamma^k_{\text{lr}} \to \mathbb{N}$ is modular in L^*.*

Proof. We prove that, for any two $C_1, C_2 \in \Gamma^2_{\text{lr}}$, the function $f'_\ell(\cdot)$ is modular in the lattice $(\Gamma^2_{\text{lr}}, \preceq^2)$, where $\preceq$ is the componentwise order of the poset L of feasible solutions. Since the sum of modular functions is modular, and d_{abs} can be written as a sum of functions $f'_\ell(\cdot)$ over all $\ell \in [r]$, the modularity of d_{abs} follows.

Let $C_1 = (X_1, X_2)$ and $C_2 = (Y_1, Y_2)$. By definition, the join ($\vee$) and meet ($\wedge$) of C_1 and C_2 are given by componentwise maximum and minimum. Then, for each $\ell \in [r]$,

$$f'_\ell(C_1 \wedge C_2) = \|\min(X_1(\ell), Y_1(\ell)) - \min(X_2(\ell), Y_2(\ell))\| \quad \text{and}$$
$$f'_\ell(C_1 \vee C_2) = \|\max(X_1(\ell), Y_1(\ell)) - \max(X_2(\ell), Y_2(\ell))\|\,.$$

Consider an arbitrary $\ell \in [r]$. Because C_1 and C_2 are each in left-right order, we have: $X_1(\ell) \preceq X_2(\ell)$ and $Y_1(\ell) \preceq Y_2(\ell)$. Consider then the intervals $I_X = [X_1(\ell), X_2(\ell)]$ and $I_Y = [Y_1(\ell), Y_2(\ell)]$. Without loss of generality, assume that $X_2(\ell) \preceq Y_2(\ell)$. There are three possibilities for the interaction of I_X and I_Y: (i) the intervals are disjoint (i.e., $I_X \cap I_Y = \emptyset$), they overlap (i.e., $I_X \cap I_Y \neq \emptyset$), or (iii) one is contained in the other (i.e., $I_X \subset I_Y$). We now compare the sums $f'_\ell(C_1 \wedge C_2) + f'_\ell(C_1 \vee C_2)$ and $f'_\ell(C_1) + f'_\ell(C_2)$ in each of these cases.

In cases (i) and (ii), we have that $X_1(\ell) \preceq Y_1(\ell)$ and $X_2(\ell) \preceq Y_2(\ell)$. Hence,

$$f'_\ell(C_1 \wedge C_2) + f'_\ell(C_1 \vee C_2) = \|X_1(\ell) - X_2(\ell)\| + \|Y_1(\ell) - Y_2(\ell)\| = f'_\ell(C_1) + f'_\ell(C_2),$$

and thus, modularity is satisfied.

In case (iii), we have $Y_1(\ell) \preceq X_1(\ell)$ and $X_2(\ell) \preceq Y_2(\ell)$. Then:

$$\begin{aligned} f'_\ell(C_1 \wedge C_2) + f'_\ell(C_1 \vee C_2) &= \|Y_1(\ell) - X_2(\ell)\| + \|X_1(\ell) - Y_2(\ell)\| \\ &= (X_2(\ell) - Y_1(\ell)) + (Y_2(\ell) - X_1(\ell)) \\ &= (X_2(\ell) - X_1(\ell)) + (Y_2(\ell) - Y_1(\ell)) \\ &= f'_\ell(C_1) + f'_\ell(C_2), \end{aligned}$$

which again satisfies modularity.

Therefore, the function $f'_\ell(\cdot)$ is modular in $(\Gamma^2_{\text{lr}}, \preceq^2)$, and the lemma is proved. ◀

By replacing the Cost-Equivalence and Submodularity lemmas of Section 3 with Lemmas 29 and 30 above, the proof of (the second half of) Theorem 28 is complete.

6 A Simple Framework for Disjoint Solutions

Finally, we consider the special case of diversity where solutions are required to be pairwise disjoint. Specifically, we consider the problem Max-Disjoint Solutions, defined below, and outline an algorithm for solving it that bypasses the need for submodular function minimization, provided that properties 1 and 2 of Theorem 1 are satisfied.

Max-Disjoint Solutions. *Given a finite set E of size n, an implicitly defined family Γ of subsets of E, referred to as feasible solutions, and a membership oracle $\mathcal{O}_\Gamma$ for Γ, find a set $C \subseteq \Gamma$ such that $X \cap Y = \emptyset$ for all $X, Y \in C$, and $|C|$ is as large as possible.*

We assume that the set Γ of feasible solutions satisfies properties 1 and 2 of Theorem 1. That is, there is a poset $P = (E, \leq)$ that is the disjoint union of r chains $(E_i, \preceq_i)$, $i \in [r]$, and the set $\Gamma \subseteq E_1 \times \cdots \times E_r$, with componentwise order $\preceq$, forms a distributive lattice.

The idea behind the algorithm is simple: start by finding the bottom element of the lattice of feasible solutions (using the $\mathcal{O}_{\min}$ oracle below), remove it along with any other solutions that overlap with it (enabled by the $\mathcal{O}_{\text{ds}}$ oracle below), and then repeat this process on the remaining sublattice until no feasible solutions remain. Of course, we want to avoid working on the lattice directly, as it can be of exponential size. Instead, we assume that the algorithm has access to the following oracles, or subroutines:

- Minimal/Maximal Solution Oracles ($\mathcal{O}_{\min}$ and $\mathcal{O}_{\max}$): On input $\langle P, \mathcal{O}_\Gamma \rangle$, the *minimal solution oracle* ($\mathcal{O}_{\min}$) returns the bottom element of the distributive lattice $(\Gamma, \preceq)$, while the *maximal solution oracle* ($\mathcal{O}_{\max}$) returns its top element; i.e.,

$$\mathcal{O}_{\min}(P, \mathcal{O}_\Gamma) = \bigwedge_{X \in \Gamma} X, \quad \text{and} \quad \mathcal{O}_{\max}(P, \mathcal{O}_\Gamma) = \bigvee_{X \in \Gamma} X.$$

- Disjoint Successors Oracle ($\mathcal{O}_{\text{ds}}$): For a feasible solution $X \in \Gamma$, the subset $\Gamma(X) \subset \Gamma$ of *disjoint successors of* X consists of all feasible solutions that are both disjoint from X and successors of X with respect to the order $\preceq$; that is, $\Gamma(X) = \{Y \mid Y \in \Gamma, X \cap Y = \emptyset, X \preceq Y\}$. Given an input $\langle X, P, \mathcal{O}_\Gamma \rangle$, this oracle returns the subposet of P induced by the subset of elements of E that appear in the disjoint succesors of X; i.e.,

$$\mathcal{O}_{\text{ds}}(X, E, \mathcal{O}_\Gamma) = P\left[\bigcup \Gamma(X)\right].$$

In this general framework, we achieve the following result.

▶ **Theorem 31.** *Max-Disjoint Solutions can be solved in $O(n)$ oracle calls.*

In other words, if subroutines $\mathcal{O}_{\min}$, $\mathcal{O}_{\max}$, and $\mathcal{O}_{\text{ds}}$ could be computed in polynomial time, we could solve Max-Disjoint Solutions in polynomial time as well. Note that these subroutines are problem-specific and must be designed and implemented based on the particular problem defined by P and Γ. This framework has been applied implicitly in prior work on disjoint minimum s-t cuts [7] and stable matchings [13], where the oracles above run in near-linear time. Our algorithm extends these ideas to a more general setting – namely, to problems satisfying properties 1 and 2 of Theorem 1.

Next, we give a formal description of the algorithm and prove its correctness and complexity, completing the proof of Theorem 31.

Preliminaries. Before we formally describe the algorithm, we require some results and notation. Throughout, let L denote the distributive lattice $(\Gamma, \preceq)$. We use X_z and X_o to denote the top and bottom elements of a lattice L, respectively, which are the two elements that satisfy $X_o \preceq X \preceq X_z$ for all $X \in \Gamma$. For a feasible solution $X \in \Gamma$, we use $X(\ell)$ to denote the element in the ℓ-th component of X. Note that $X(\ell) \in E_\ell$. The following observation is a necessary condition for the existence of disjoint solutions in Γ.

▶ **Observation 32.** *Let $e \in E$ be an element of both X_o and X_z. Then e must be present in every feasible solution in Γ.*

Proof. Consider an arbitrary feasible solution $X \in \Gamma$. Without loss of generality, let $e \in E_\ell$. By definition of bottom element of L, we have $X_o \preceq X$, and thus $e \preceq_\ell X(\ell)$. On the other hand, by definition of top element of L, we have $X \preceq X_z$, which implies $X(\ell) \preceq_\ell e$. By antisymmetry of the partial order $\preceq_\ell$, it follows that $X(\ell) = e$. Hence, $e \in X$, proving the fact. ◀

We also make the following observation about the set of disjoint successors of a feasible solution. With a slight abuse of notation, we use $\preceq$ to denote to the componentwise ordering arising from P and any induced suposet.

▶ **Observation 33.** *For any $X \in \Gamma$, the set $\Gamma(X)$ of disjoint successors of X satisfies properties 1 and 2 of theorem 1.*

Proof. We start with the first property. Let $P(X) = [\bigcup \Gamma(X)]$ denote the subposet induced by $\bigcup \Gamma(X)$. This subposet then consists of the disjoint chains of P but restricted to the elements appearing in $\bigcup \Gamma(X)$. By definition of both $P(X)$ and $\Gamma(X)$, each solution in $\Gamma(X)$ must contain exactly one element from each chain in $P(X)$. Hence, property 1 is satisfied.

As for the second property, it is clear that $\Gamma(X) \subset \Gamma$. Moreover, because the join ($\vee$) and meet ($\wedge$) operations in L are defined as the componentwise maximum and minimum, respectively, $\Gamma(X)$ remains closed under these operations. This means that the poset $(\Gamma(X), \preceq)$ is a sublattice of L and thus, a distributive lattice. Hence, property 2 is satisfied. ◀

With these results, we are ready to describe and analyze the algorithm.

The algorithm. Given an input $\langle P, \mathcal{O}_\Gamma \rangle$, the algorithm begins by determining the bottom element X_o and the top element X_z of lattice L by querying the oracles $\mathcal{O}_{\min}$ and $\mathcal{O}_{\max}$ with the input $\langle P, \mathcal{O}_\Gamma \rangle$. If these two solutions share an element, the algorithm stops, as Observation 32 ensures that no disjoint solutions exist. Otherwise, it proceeds by querying $\mathcal{O}_{\text{ds}}(X_o, P, \mathcal{O}_\Gamma)$ to determine the subposet $P(X_o)$ induced by $\bigcup \Gamma(X_o)$.

By Observation 33, the set $\Gamma(X_o)$ satisfies properties 1 and 2 of Theorem 1, with the poset $P(X_o)$ serving as the corresponding chain decomposition. Let $L(X_o) = (\Gamma(X_o), \preceq)$ be the associated sublattice of disjoint successors of X_o. The algorithm proceeds by querying $\mathcal{O}_{\min}$ with the input $\langle P(X_o), \mathcal{O}_\Gamma \rangle$ to identify the bottom element X'_o of $L(X_o)$. Once more, if X'_o is disjoint from X_z, the algorithm queries $\mathcal{O}_{\text{ds}}(X'_o, P(X_o), \mathcal{O}_\Gamma)$ to determine the subposet $P(X'_o)$ induced by the set $\bigcup \Gamma(X'_o)$ of disjoint successors of X'_o. This process repeats as long as $\mathcal{O}_{\min}$ continues to return solutions that are disjoint from X_z. Throughout the execution, the algorithm maintains a set C that stores all solutions found that are disjoint from X_z and returns this set upon termination. The algorithm is presented below as Algorithm 1.

Correctness. The solutions in the set $C = \{X_1, X_2, \ldots, X_k\}$ returned by Algorithm 1 are clearly disjoint by construction, as the poset returned by the oracle $\mathcal{O}_{\text{ds}}$ at each step is induced by the set of disjoint successors of the solution identified in the precious step. Moreover, the set C is, in fact, a left-right ordered tuple. This follows again by construction, as each newly

Algorithm 1 MAX-DISJOINT SOLUTIONS.

Input: A poset P and a membership oracle $\mathcal{O}_\Gamma$ satisfying properties 1 and 2 of Theorem 1.
Output: A maximum cardinality set C of disjoint feasible solutions from Γ.

$C \leftarrow \emptyset$
$X_z \leftarrow \mathcal{O}_{\max}(P, \mathcal{O}_\Gamma)$ ▷ Top element of lattice L.
$X \leftarrow \mathcal{O}_{\min}(P, \mathcal{O}_\Gamma)$ ▷ Bottom element of lattice L.
$P(X) \leftarrow \mathcal{O}_{\mathrm{ds}}(P, X_o, \mathcal{O}_\Gamma)$ ▷ This defines a new instance.
while $X \cap X_z = \emptyset$ **do**
  $C \leftarrow C \cup \{X\}$
  $X \leftarrow \mathcal{O}_{\min}(P(X), \mathcal{O}_\Gamma)$ ▷ New disjoint solution.
  $P(X) \leftarrow \mathcal{O}_{\mathrm{ds}}(P, X, \mathcal{O}_\Gamma)$
$C \leftarrow C \cup \{X\}$
return C

identified solution is determined from the subset of elements that are chain-successors of elements included in previously identified solutions. Note that the notion of left-right order here is strict, meaning that $X_i \prec X_j$ for any $1 \le i < j \le k$.

To analyze this further, let us go back for a moment to Section 3. Note that the d_{sum} measure is maximum whenever its input consists of disjoint solutions. Consider then an arbitrary k-tuple of disjoint feasible solutions, for some $k > 0$. We know, by Claim 6, that there exists a k-tuple of disjoint feasible solutions that is in left-right order. In particular, this is true for a disjoint-solutions tuple of maximum cardinality k^*. Then, as we did in Section 3, we may restrict our arguments to the set of k^*-tuples that are in left-right order without loss of generality.

To complete the correctness of Algorithm 1, it remains to show that the tuple returned by the algorithm is of maximum cardinality k^*.

▶ **Lemma 34.** *Algorithm 1 outputs a longest tuple of disjoint feasible solutions.*

Proof. Let $C_{\mathrm{ALG}} = (X_1, X_2, \dots, X_k)$ be the k-tuple of disjoint feasible solutions returned by Algorithm 1. For the sake of contradiction, suppose that $C' = (Y_1, Y_2, \dots, Y_\ell)$ is a longest left-right ordered tuple of disjoint feasible solutions with $\ell > k$.

By definition of bottom element, we know that solution $X_1 = X_o$ is a predecessor of every other feasible solution in Γ. This implies that $Y_1 \cap X_1 \neq \emptyset$; otherwise, we could append X_1 to the start of C' and obtain a longer tuple of left-right ordered disjoint solutions. Then, we have $X_1 \preceq Y_1 \prec Y_2$, and we may replace Y_1 in C' with X_1 to generate a new ℓ-tuple C_1 of disjoint solutions.

By Observation 33, and the definition of bottom element, we know that solution X_2 found by the algorithm is a predecessor of every feasible solution in $\Gamma(X_1)$; that is, X_2 is a predecessor of every feasible solution disjoint from X_1. By the same argument as before, $X_2 \cap Y_2 \neq \emptyset$. We then have $X_2 \preceq Y_2 \prec Y_3$, and we may replace Y_2 in C_1 with X_2 to generate a new ℓ-tuple C_2 of disjoint solutions.

By repeating this procedure k times, we end up with the ℓ-tuple $C_k = (X_1, X_2, \dots, X_k, Y_{k+1}, \dots, Y_\ell)$ of left-right ordered disjoint solutions. Then, there exists a feasible solution Y_{k+1} that is a strict successor of X_k – the last element of tuple C_{ALG}. But this implies that X_k is disjoint with the top element X_z of L, which we know to be false by construction of C_{ALG}. Thus, we get the necessary contradiction. ◀

Time complexity. The oracles $\mathcal{O}_{\min}$ and $\mathcal{O}_{\mathrm{ds}}$ are called k^* times, and k^* is upper bounded by the length of the shortest chain in P, which, in the worst case, has length $O(n)$. This completes the proof of Theorem 31.

7 Concluding Remarks

We showed that Max-Cov k-Diverse Solutions can be solved in polynomial time by reducing it to submodular function minimization on a distributive lattice, provided the feasible solution set satisfies three structural properties. This gives a general framework for designing polynomial-time algorithms for diverse variants of combinatorial problems. We applied it to Minimum s-t Cut, Stable Matching, and Market Clearing Price, and showed it extends beyond the pairwise-sum measure to the coverage and absolute-difference measures. We also showed that in the special case where diversity is defined by pairwise disjointness, the problem can be solved without relying on submodular function minimization.

A natural direction for future work is identifying more problems that satisfy the structural properties of Theorem 1, and to characterize which diversity measures are compatible with the framework. It remains open whether the reliance on SFM can be avoided without losing generality.

References

1. Julien Baste, Lars Jaffke, Tomáš Masařík, Geevarghese Philip, and Günter Rote. Fpt algorithms for diverse collections of hitting sets. *Algorithms*, 12(12):254, 2019. doi:10.3390/A12120254.
2. Garrett Birkhoff. Rings of sets. *Duke Mathematical Journal*, 3(3):443–454, 1937.
3. Charles Blair. The lattice structure of the set of stable matchings with multiple partners. *Mathematics of operations research*, 13(4):619–628, 1988. doi:10.1287/MOOR.13.4.619.
4. Mohammadreza Bolandnazar, Woonghee Tim Huh, S Thomas McCORMICK, and Kazuo Murota. A note on "order-based cost optimization in assemble-to-order systems". *University of Tokyo (February, Techical report*, 2015.
5. Paul Bonsma. Most balanced minimum cuts. *Discrete Applied Mathematics*, 158(4):261–276, 2010. doi:10.1016/J.DAM.2009.09.010.
6. Brian A Davey and Hilary A Priestley. *Introduction to lattices and order*. Cambridge university press, 2002.
7. Mark de Berg, Andrés López Martínez, and Frits Spieksma. Finding diverse minimum st cuts. In *34th International Symposium on Algorithms and Computation*, 2023.
8. Mark de Berg, Andrés López Martínez, and Frits Spieksma. Finding diverse solutions in combinatorial problems with a distributive lattice structure, 2025. doi:10.48550/arXiv.2504.02369.
9. Karolina Drabik and Tomáš Masařík. Finding diverse solutions parameterized by cliquewidth. *arXiv preprint*, 2024. arXiv:2405.20931.
10. David Easley, Jon Kleinberg, et al. *Networks, crowds, and markets: Reasoning about a highly connected world*, volume 1. Cambridge university press Cambridge, 2010.
11. Fernando Escalante. Schnittverbände in graphen. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 38, pages 199–220. Springer, 1972.
12. Fedor V Fomin, Petr A Golovach, Lars Jaffke, Geevarghese Philip, and Danil Sagunov. Diverse pairs of matchings. *Algorithmica*, 86(6):2026–2040, 2024. doi:10.1007/S00453-024-01214-7.
13. Aadityan Ganesh, HV Vishwa Prakash, Prajakta Nimbhorkar, and Geevarghese Philip. Disjoint stable matchings in linear time. In *Graph-Theoretic Concepts in Computer Science: 47th International Workshop, WG 2021, Warsaw, Poland, June 23–25, 2021, Revised Selected Papers 47*, pages 94–105. Springer, 2021. doi:10.1007/978-3-030-86838-3_7.

14 Rohith Reddy Gangam, Tung Mai, Nitya Raju, and Vijay V Vazirani. A structural and algorithmic study of stable matching lattices of "nearby" instances, with applications. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.FSTTCS.2022.19.

15 Vijay K Garg. *Introduction to lattice theory with computer science applications*. John Wiley & Sons, 2015.

16 Vijay K Garg. Applying predicate detection to the constrained optimization problems. *arXiv preprint*, 2018. arXiv:1812.10431.

17 Vijay K Garg. Predicate detection to solve combinatorial optimization problems. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 235–245, 2020. doi:10.1145/3350755.3400235.

18 Vijay K Garg and Neeraj Mittal. On slicing a distributed computation. In *Proceedings 21st International Conference on Distributed Computing Systems*, pages 322–329. IEEE, 2001. doi:10.1109/ICDSC.2001.918962.

19 George Gratzer. *Lattice theory: First concepts and distributive lattices*. Courier Corporation, 2009.

20 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

21 D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. Foundations of computing. MIT Press, 1989. URL: https://books.google.nl/books?id=2TzhSQAACAAJ.

22 R Halin. Lattices related to separation in graphs. In *Finite and Infinite Combinatorics in Sets and Logic*, pages 153–167. Springer, 1993.

23 Tesshu Hanaka, Masashi Kiyomi, Yasuaki Kobayashi, Yusuke Kobayashi, Kazuhiro Kurita, and Yota Otachi. A framework to design approximation algorithms for finding diverse solutions in combinatorial problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3968–3976, 2023. doi:10.1609/AAAI.V37I4.25511.

24 Tesshu Hanaka, Yasuaki Kobayashi, Kazuhiro Kurita, and Yota Otachi. Finding diverse trees, paths, and more. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3778–3786, 2021. doi:10.1609/AAAI.V35I5.16495.

25 Egbert Harzheim. *Ordered sets*, volume 7. Springer Science & Business Media, 2005.

26 Yuni Iwamasa, Tomoki Matsuda, Shunya Morihira, and Hanna Sumita. A general framework for finding diverse solutions via network flow and its applications. *arXiv preprint*, 2025. doi:10.48550/arXiv.2504.17633.

27 Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001. doi:10.1145/502090.502096.

28 Donald Ervin Knuth. *Stable marriage and its relation to other combinatorial problems: An introduction to the mathematical analysis of algorithms*, volume 10. American Mathematical Soc., 1997.

29 Soh Kumabe. Max-distance sparsification for diversification and clustering. *arXiv preprint*, 2024. doi:10.48550/arXiv.2411.02845.

30 George Markowsky. An overview of the poset of irreducibles. *Combinatorial And Computational Mathematics*, pages 162–177, 2001.

31 Bernd Meyer. On the lattices of cutsets in finite graphs. *European Journal of Combinatorics*, 3(2):153–157, 1982. doi:10.1016/S0195-6698(82)80028-0.

32 Neeldhara Misra, Harshil Mittal, and Ashutosh Rai. On the parameterized complexity of diverse sat. In *35th International Symposium on Algorithms and Computation (ISAAC 2024)*, pages 50:1–50:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ISAAC.2024.50.

33 Neeraj Mittal and Vijay K Garg. Computation slicing: Techniques and theory. In *Distributed Computing: 15th International Conference, DISC 2001 Lisbon, Portugal, October 3–5, 2001 Proceedings 15*, pages 78–92. Springer, 2001. doi:10.1007/3-540-45414-4_6.

34 Kazuo Murota. *Discrete Convex Analysis*. Society for Industrial and Applied Mathematics, 2003. doi:10.1137/1.9780898718508.

35 Jean-Claude Picard and Maurice Queyranne. On the structure of all minimum cuts in a network and applications. *Math. Program.*, 22(1):121, December 1982. doi:10.1007/BF01581031.

36 Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000. doi:10.1006/JCTB.2000.1989.

37 Lloyd S Shapley and Martin Shubik. The assignment game I: The core. *International Journal of game theory*, 1(1):111–130, 1971.

38 Yuto Shida, Giulia Punzi, Yasuaki Kobayashi, Takeaki Uno, and Hiroki Arimura. Finding diverse strings and longest common subsequences in a graph. In *35th Annual Symposium on Combinatorial Pattern Matching (CPM 2024)*, pages 27:1–27:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CPM.2024.27.

39 Marilda Sotomayor. The lattice structure of the set of stable outcomes of the multiple partners assignment game. *International Journal of Game Theory*, 28:567–583, 1999. doi:10.1007/S001820050126.

40 Richard P Stanley. Enumerative combinatorics: Volume 1, 2011.