

Scalable Enumeration of Pareto-Optimal Polymers for Computing Equilibrium Concentrations

Archit Patil ✉ 

University of Texas at Austin, TX, USA

Minki Hhan ✉ 

Korea Advanced Institute of Science and Technology, Daejeon, South Korea

David Soloveichik ✉ 

University of Texas at Austin, TX, USA

Abstract

Predicting equilibrium concentrations of molecular complexes is essential for verifying the behavior of engineered DNA systems. However, a finite set of monomer types can in principle generate infinitely many complexes. We study this candidate-enumeration problem in a geometry-free, domain-level abstraction called a domain-monomer system, generalizing Thermodynamic Binding Networks (TBNs) to the unsaturated setting where not every possible bond need be formed. We define *Pareto-suboptimal* polymers as those that can be split into non-interacting parts, and show that restricting attention to *Pareto-optimal* polymers is thermodynamically justified: no Pareto-suboptimal polymer appears in any minimum free-energy configuration, and the total equilibrium concentration of such polymers is small. We prove that there are finitely many Pareto-optimal polymers and exactly characterize them via a Hilbert basis computation, extending prior work from the saturated TBN model. To scale this approach to large systems, we develop a framework that restricts the number of different monomer types that a single polymer contains, and uses combinatorial covering designs to reduce the number of Hilbert basis computations required. We benchmark the method on several families of DNA molecular programming systems, demonstrating order-of-magnitude speedups over direct computation while recovering nearly all equilibrium-relevant polymers.

2012 ACM Subject Classification Computer systems organization → Molecular computing; Theory of computation → Design and analysis of algorithms; Applied computing → Computational biology

Keywords and phrases Molecular computation, Hilbert Basis, Thermodynamic Binding Network, covering design, equilibrium concentration

Digital Object Identifier 10.4230/LIPIcs.DNA.32.12

Supplementary Material *Software (Github Repo)*: <https://github.com/architrahul/Pareto-polymer-enumerator>; archived at `swh:1:dir:b5917c1073ebd7f44f1d3c4b5420671254195cee`
Text (Supplementary Text): https://github.com/architrahul/Pareto-polymer-enumerator/blob/main/Supplementary_Text.pdf

Funding Schmidt Sciences Polymath Award to D.S., DOE grant DE-SC0024248, NSF SemiSynBio III: GOALI grant 2227578.

Acknowledgements We thank And Kaan Ata Yilmaz for valuable discussions and insights.

1 Introduction

Understanding and engineering complex molecular systems and behaviors requires both kinetic and thermodynamic modeling. In the context of dynamic DNA nanotechnology, NUPACK [10, 12] is widely used to analyze the minimum free energy configurations of complexes, and their concentrations at thermodynamic equilibrium. However, since NUPACK operates at the individual base-pair level, it is infeasible to use for large systems (many strands and/or large complex sizes). At the same time, molecular programming researchers have



© Archit Patil, Minki Hhan, and David Soloveichik;
licensed under Creative Commons License CC-BY 4.0

32nd International Conference on DNA Computing and Molecular Programming (DNA 32).

Editors: Dominic Scalise and Robert Schweller; Article No. 12; pp. 12:1–12:24

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

been developing systems of ever larger complexity. For example, recent work demonstrating DNA neural networks and logic circuits involved 700 distinct complexes and over 200 strands in a single test tube [7, 20]. Such complexity necessitates a higher level of abstraction.

Most engineered DNA systems are organized around domains, contiguous sequence regions that behave as logical binding units. Domain-level abstractions make different geometric commitments: some models enforce rigid geometry (e.g., tile assembly), while others ignore geometry and retain only which domain types can bind. In this paper we study the latter, geometry-free setting. Our formalism is a *domain-monomer system*, closely related to Thermodynamic Binding Networks (TBNs) [11, 4, 3, 15, 22], in which a monomer is a multiset of abstract domains, complementary domains can bind, and a polymer is a multiset of monomers. Unlike the saturated TBN setting which is meant to capture strong bonding, we do not require every possible bond to be formed. The solution might have polymers with unbound complementary domains, corresponding to a more realistic model that permits bonds of varying affinities. The central thermodynamic question is to determine the equilibrium distribution of polymer concentrations: which polymers form, and in what amounts? In engineered systems, some polymers are intended products and verifying correct behavior requires checking their equilibrium concentrations (e.g., that certain polymers are present in specific ratios), while others are undesired leakage.

In typical systems there is a relatively small set of polymers that dominate the equilibrium, with the rest present in negligible concentrations (a finite set of monomer types can generate infinitely many polymers). If we could a priori eliminate the vast majority of safe-to-neglect polymers, then equilibrium optimization itself would be efficient (tools such as COFFEE [24] can handle hundreds of thousands of candidate polymers). But constructing a sensible finite candidate set of polymers in a principled way appears nontrivial as it must be defined independently of the unknown equilibrium solution itself. We address this difficulty with a natural structural notion called *Pareto-optimality*. Informally, a polymer is Pareto-suboptimal if it can be split into two non-complementary polymers. More precisely, a polymer is Pareto-suboptimal if it is possible to reconfigure the monomers of the polymer into two or more separate polymers without decreasing overall bonding. Such a split preserves the total bond energy (enthalpy) while producing more separate complexes (increasing entropy), so Pareto-suboptimal polymers are thermodynamically disfavored. We formally justify the focus on Pareto-optimal polymers by showing that no Pareto-suboptimal polymer appears in a minimum free-energy configuration in a discrete model, and further that in the continuous equilibrium model their total concentration is bounded in terms of the total concentration of Pareto-optimal polymers. This makes Pareto-optimal polymers a principled candidate set for equilibrium analysis.

Prior work of Haley and Doty [15] connected Hilbert bases to polymer enumeration in the saturated TBN model, which takes the thermodynamic limit of strong bonds. Our goal is to extend this algebraic perspective to the more general unsaturated setting of possibly weak bonds, allowing different bond strengths corresponding to hybridization energies of different domains. Further, we want to make the analysis practical on large systems where direct Hilbert basis computation is far too expensive.

Our contributions are as follows.

- We bound the total concentration of Pareto-suboptimal polymers in geometry-free domain-monomer systems, justifying their exclusion.
- We show that the full set P^* of Pareto-optimal polymers is finite and can be characterized exactly via Hilbert basis enumeration. This extends the Hilbert-basis connection from the saturated TBN, discrete setting [15].

- We develop a scalable framework for enumerating the subset $P_t^* \subseteq P^*$ of Pareto-optimal polymers that use at most t monomer types (*support-boundedness restriction*). We first express P_t^* using Hilbert bases of smaller subsystems, and reduce the number of required subsystem computations by using covering designs. The resulting method returns a superset $\hat{P}_{k,t}$ satisfying $P_t^* \subseteq \hat{P}_{k,t} \subseteq P^*$. Importantly, the parameter t is interpretable: rather than imposing an opaque algebraic or numerical cutoff, it bounds the number of distinct monomer types in a polymer, a quantity often apparent from the design specification of the intended products and plausible leakage pathways.
- We describe implementation techniques, including a probe-and-prune heuristic for choosing the block size k , and benchmark the method on several families of molecular systems from the recent literature [22, 23, 21]. For example, on a 7-module linear cascade, the covering-design strategy with $(t, k) = (5, 25)$ reduces polymer enumeration time from >1000 s for direct full Hilbert basis computation to 24 s, without missing any equilibrium-relevant polymer in the corresponding leakage analysis.

Our results provide one theoretical and algorithmic answer to the candidate-enumeration problem for equilibrium analysis in large geometry-free molecular systems: Pareto-optimal polymers give a well-justified reduced search space characterized by a Hilbert basis enumeration, and covering-design-based support restriction makes computation feasible in practice.

2 Preliminaries

2.1 Model

Let $\mathbb{Z}, \mathbb{N}, \mathbb{R}$, and $\mathbb{R}_{\geq 0}$ denote the sets of integers, nonnegative integers, real numbers, and nonnegative real numbers, respectively. For a finite set A , we define $\mathbb{N}^A, \mathbb{Z}^A, \mathbb{R}^A$, and $\mathbb{R}_{\geq 0}^A$ as the sets of $|A|$ -dimensional vectors over $\mathbb{N}, \mathbb{Z}, \mathbb{R}$, and $\mathbb{R}_{\geq 0}$, respectively. The vector $\mathbf{x} \in \mathbb{N}^A$ (or $\mathbb{Z}^A, \mathbb{R}^A, \mathbb{R}_{\geq 0}^A$) is represented by $(\mathbf{x}(a))_{a \in A}$, or sometimes $(\mathbf{x}_a)_{a \in A}$ if there is no confusion. A multiset $\mathbf{x}$ over A is an unordered collection of elements allowing duplicates, which is naturally identified by a vector $\mathbf{x}$ in $\mathbb{N}^A$. Similarly, we define $\mathbb{Z}^{A \times B}$ as the set of $|A| \times |B|$ -matrices over $\mathbb{Z}$ for finite sets A, B . The matrix-vector multiplication $M\mathbf{x}$ for $M \in \mathbb{Z}^{A \times B}$ and $\mathbf{x} \in \mathbb{Z}^B$ (or $\mathbb{N}^B, \mathbb{R}^B, \mathbb{R}_{\geq 0}^B$) is defined in a natural way.

We use the following three-level abstraction motivated by DNA nanotechnology and the TBN model [11, 4]. This model captures how monomers (indivisible molecules) are described by domains, and how they combine to form polymers (complexes).

- **Definition 1.** A domain-monomer system is a pair (Σ, M) where
- $\Sigma = D \cup D^*$ is a set of domains described by D and D^* , which are the disjoint and same-sized sets of primal and complementary domains, respectively. For each $a \in D$, there exists a corresponding $a^* \in D^*$ that can form a bond with a , i.e., $D^* = \{a^* : a \in D\}$. We say a and a^* are complementary to each other, and call (a, a^*) the domain pair.
 - M is a finite set of monomers, each of which is a finite multiset of domains over Σ .

Each monomer $m \in M$ is encoded as a vector $\mathbf{a}_m \in \mathbb{Z}^D$, where the entry corresponding to the domain pair $\sigma = (a, a^*)$ is $+1$ for each occurrence of a in m and -1 for a^* . This gives the monomer matrix $A \in \mathbb{Z}^{D \times M}$, whose columns $\mathbf{a}_m$ correspond to each $m \in M$.

Given a domain-monomer system (Σ, M) , a polymer is defined as a finite, nonzero multiset of monomers. We represent each polymer as a vector $\mathbf{p} \in \mathbb{N}^M \setminus \{\mathbf{0}\}$ where $\mathbf{p}(m)$ (or sometimes $\mathbf{p}_m$) denotes the number of copies of monomer m in polymer $\mathbf{p}$. In other words, $P = \mathbb{N}^M \setminus \{\mathbf{0}\}$ defines the set of all possible polymers. The net domain vector of $\mathbf{p}$ is $A\mathbf{p} \in \mathbb{Z}^D$, whose entry for domain a gives the net count of a minus a^* across all monomers in the polymer. We call an equation $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ a split (of $\mathbf{p}$).

Intrapolymer bonding is assumed throughout: complementary domains within a single polymer are taken to be already bound. Under this assumption, the free domains of a polymer $\mathbf{p}$ are precisely those whose corresponding entry in $A\mathbf{p}$ is nonzero. Note that as in the TBN model, there is no underlying notion of geometry which can prevent two complementary domains within a polymer from binding.

2.2 Computing Polymer Concentrations at Equilibrium

The *concentration* $[\mathbf{p}]$ of polymer $\mathbf{p} \in P$ specifies the amount of polymer $\mathbf{p}$ in the system. In our theoretical results, we use dimensionless mole fractions [10], capturing the amount of the polymer relative to the solvent molecules (water). In normal dilute solutions, $\sum_{\mathbf{p} \in P} [\mathbf{p}] \ll 1$. For each $m \in M$, the polymer concentration induces the total monomer concentration

$$[m] = \sum_{\mathbf{p} \in P} \mathbf{p}_m [\mathbf{p}]. \quad (1)$$

Equilibrium analysis aims to determine the concentrations of polymers at equilibrium, given an initial supply of monomers (or polymers). NUPACK’s `complex_concentrations` function [12]¹ and COFFEE [24] can compute equilibrium concentrations given the initial monomer concentrations and a candidate set of polymers. However, the main computational bottleneck is typically not the equilibrium computation itself, but the construction of an appropriate candidate set of polymers. For a given domain-monomer system, the set of possible polymers is generally infinite and cannot be enumerated exhaustively.

In this paper, we introduce the principled restriction to polymers that are Pareto-optimal. As we show next, Pareto-suboptimal polymers can be excluded on thermodynamic grounds. This reduces the search space from an intractable infinite family to a finite set of candidates, after which additional enumeration strategies can be applied.

For complex systems, even restricting to Pareto-optimal polymers is computationally intractable. This motivates our subsequent methodology of limiting the “size” of the enumerated Pareto-optimal polymers, where size can be defined in two ways: (1) the number of different types of monomers, or (2) the number of types of domain pairs in the polymer.

► **Remark 2.** A domain-monomer system corresponds exactly to a Thermodynamic Binding Network (TBN) [11, 4] under two relaxed assumptions. First, saturation is not assumed: the net domain vector $A\mathbf{x}$ of a polymer $\mathbf{x}$ need not be nonnegative, meaning unstarred domains may remain unbound. Second, no assumption is made that enthalpy dominates entropy. The present work analyzes TBNs from this more general perspective, without either idealization.

3 Pareto-Optimal Polymers

3.1 Defining Pareto-Optimal Polymers

Throughout this paper, we fix the domain-monomer system (Σ, M) unless specified otherwise. In this section, we introduce reasoning that allows us to exclude certain polymers from consideration, based on the observation that they do not occur in minimum free energy configurations. This leads to a *finite* search space instead of the infinite set of all polymers.

► **Definition 3** (Complementary polymers). *Two polymers $\mathbf{p}, \mathbf{q} \in P$ are complementary if there exists a domain such that the entries of $A\mathbf{p}$ and $A\mathbf{q}$ corresponding to that domain are both nonzero and have opposite sign. In other words, $(A\mathbf{p})_a \cdot (A\mathbf{q})_a < 0$ for some $a \in D$.*

¹ Equivalently the `concentrations` command line tool for older NUPACK 3.

Equivalently, $\mathbf{p}$ and $\mathbf{q}$ are complementary if and only if the combined polymer $\mathbf{p} + \mathbf{q}$ contains at least one intrapolymer bond between a domain of $\mathbf{p}$ and a domain of $\mathbf{q}$, so that splitting $\mathbf{p} + \mathbf{q}$ back into $\mathbf{p}$ and $\mathbf{q}$ necessarily breaks at least one bond.

► **Definition 4** (Pareto-(sub)optimal polymer). *A polymer $\mathbf{p} \in P$ is Pareto-suboptimal if there exists a split $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ with $\mathbf{p}_1, \mathbf{p}_2 \in P$ such that $\mathbf{p}_1$ and $\mathbf{p}_2$ are not complementary. If $\mathbf{p}$ is not Pareto-suboptimal, it is Pareto-optimal. The set of all Pareto-optimal polymers is denoted P^* .*

Figure 1a illustrates a Pareto-suboptimal polymer: its monomers can be regrouped into two non-complementary polymers without breaking any bond. We stress that the model does not track bonds between specific domains. A polymer is represented only by its monomer multiset $\mathbf{p}$; the explicit bonds drawn in the figure are an informal visualization rather than part of the model. From the model's perspective, the reconfiguration changes only how the monomers are partitioned into separate polymers, leaving the bonding unchanged. Note that the geometry-free nature of the model is essential here. Geometric constraints could permit $\mathbf{p}$ but not its decomposition into $\mathbf{p}_1 + \mathbf{p}_2$.

3.2 Thermodynamic Justification

In this section we justify the exclusion of Pareto-suboptimal polymers on thermodynamic grounds, arguing that they are in a certain sense negligible. We begin with the discrete model, showing that no Pareto-suboptimal polymer appears in any minimum free energy configuration. This provides strong motivation for the exclusion, but the gap between discrete minimum free energy configurations and continuous equilibrium concentrations calls for a quantitative bound in the continuous regime. Accordingly, in the second part of this section we derive a general bound on the total concentration of suboptimal polymers in terms of the total concentration of optimal ones.

3.2.1 Discrete thermodynamics

Towards justifying the exclusion of Pareto-suboptimal polymers, in this section we use a simple discrete model to show that any minimum free energy configuration contains only Pareto-optimal ones. Given an initial composition, i.e., a multiset of monomers, a *configuration* is a multiset of polymers built from this composition. The *free energy* of a configuration C is

$$G(C) = a \cdot H(C) - b \cdot S(C)$$

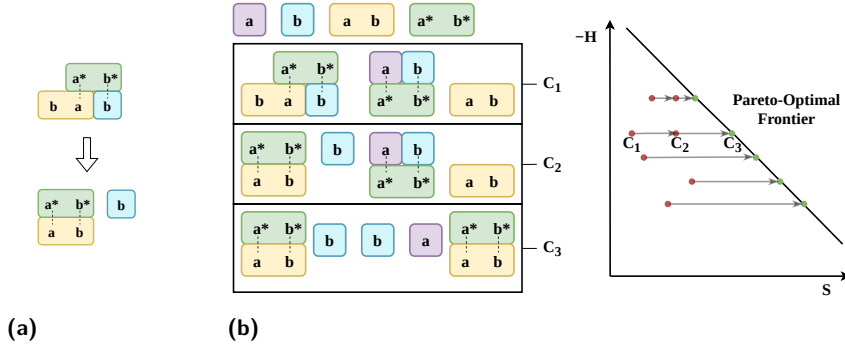
where $H(C) < 0$ is the total enthalpy (sum of bond energies across all bonds in C), $S(C)$ is the entropy (number of separate polymer units), and $a, b > 0$ are system-dependent weights. The probability of observing a configuration C is proportional to its Boltzmann weight:

$$\mathbb{P}[C] \propto e^{-G(C)/kT}$$

► **Lemma 5.** *No Pareto-suboptimal polymer appears in any minimum free energy configuration.*

The proof is deferred to Appendix A.2.

Figure 1b shows an example configuration C_1 where the first polymer is Pareto-suboptimal. Splitting this polymer yields configuration C_2 , which strictly improves on the free energy. Note that a configuration may have no Pareto-suboptimal polymer and still not lie on the Pareto-optimal frontier. For example, the second and third polymers in C_2 can be reconfigured into three polymers, which again strictly improves on the free energy (configuration C_3).



■ **Figure 1** Pareto-suboptimal polymers and the Pareto-optimal frontier. **(a)** Visualization of a Pareto-suboptimal polymer (top) whose monomers can be regrouped into two non-complementary polymers with the same number of bonds (bottom). **(b)** The Pareto-optimal frontier in the $(S, -H)$ plane. Minimum free energy configurations lie on the frontier. Configurations containing Pareto-suboptimal polymers lie strictly below the frontier; each such configuration is dominated by a configuration with the same enthalpy and strictly higher entropy (rightward arrows). While suboptimal polymers necessarily make a configuration suboptimal, the implication does not go the other way. For example, C_2 does not contain a Pareto-suboptimal polymer, but reconfiguring its monomers yields the configuration C_3 with one more polymer but the same bonds.

3.2.2 Continuous thermodynamics

Toward further justifying the exclusion of Pareto-suboptimal polymers, we now turn to the continuous concentrations regime. We show that the total concentration of Pareto-suboptimal polymers is much smaller than the total concentration of Pareto-optimal polymers at equilibrium.

The equilibrium in the continuous model corresponds to the minimizer of the pseudo-Helmholtz free energy

$$g = \sum_{\mathbf{p} \in P} [\mathbf{p}] (\log[\mathbf{p}] + G_{\mathbf{p}} - 1),$$

where $G_{\mathbf{p}} = -\log \Omega_{\mathbf{p}}$ is the formation energy of the polymer $\mathbf{p} \in P$, subject to the monomer conservation constraints Equation (1).

At equilibrium, for any (nontrivial) split $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ where $\mathbf{p}_1$ and $\mathbf{p}_2$ are not complementary, the following holds for the dimensionless mole fractions in Equation (1):

$$[\mathbf{p}] = [\mathbf{p}_1] \cdot [\mathbf{p}_2]. \quad (2)$$

This equality follows from the first-order optimality conditions of the pseudo-Helmholtz free energy minimization, which yield $[\mathbf{p}] = \Omega_{\mathbf{p}} \prod_{m \in M} z_m^{\mathbf{p}(m)}$ for some $z_m > 0$. The factorization over a non-complementary split then holds because the formation energy is additive across non-interacting parts. This is a standard result in the polymer equilibrium literature [10].

► **Theorem 6.** *Let $[\cdot]$ denote equilibrium concentrations, and define $C_{\text{opt}} := \sum_{\mathbf{p} \in P^*} [\mathbf{p}]$, and $C_{\text{sub}} := \sum_{\mathbf{p} \in P \setminus P^*} [\mathbf{p}]$. If $C_{\text{opt}} < 1$, then*

$$C_{\text{sub}} \leq \frac{C_{\text{opt}}^2}{1 - C_{\text{opt}}}.$$

In particular, in the dilute regime when $C_{\text{opt}} \ll 1$, we have $C_{\text{sub}} = O(C_{\text{opt}}^2)$, so the total concentration of Pareto-suboptimal polymers is negligible compared with the total concentration of Pareto-optimal polymers. We defer the proof to Appendix A.1.

4 Enumeration of Pareto-Optimal Polymers via Hilbert Basis Computation

This section shows that the Pareto-optimal polymers of a domain-monomer system can be identified by a Hilbert basis of a set of neutral polymers having zero net domain vectors in an extended system.

Our Hilbert-basis characterization generalizes the saturated-TBN framework of Haley and Doty [15]. Specifically, while the original TBN model [11, 4] used by the prior work focuses on configurations with maximum binding (called saturated configurations), our domain-monomer system relaxes this assumption and allows unpaired complementary domains in different polymers to coexist in a configuration. In order to make this generalization, we utilize an additional augmented neutralization construction described below.

Given a domain-monomer system (Σ, M) with monomer matrix $A \in \mathbb{Z}^{|D| \times |M|}$, we define an *augmented (domain-monomer) system* by introducing two *unit monomers* per domain pair. More precisely, the augmented system (Σ, M') is a domain-monomer system defined as follows: It inherits the original domain Σ , and M' is defined as a superset of M by

$$M' = M \cup \{u_a = \{a\}, u_{a^*} = \{a^*\} : (a, a^*) \text{ is a domain pair}\}.$$

In words, for each domain pair (a, a^*) , the unit monomer u_a contains only a , and the unit monomer u_{a^*} contains only a^* . Both are added as monomers in the augmented system. Their columns in the augmented monomer matrix are $+1$ and -1 , respectively, in the coordinates corresponding to (a, a^*) and 0 elsewhere. Let M' denote the augmented monomer set and $A' \in \mathbb{Z}^{D \times M'}$ the corresponding augmented monomer matrix. It is worth noting that for $\mathbf{w} = (\mathbf{w}_M, \mathbf{w}_\Sigma) \in \mathbb{Z}^M \times \mathbb{Z}^\Sigma = \mathbb{Z}^{M'}$, it holds that for $a \in D$

$$(A'\mathbf{w})_a = (A\mathbf{w}_M)_a + (\mathbf{w}_\Sigma)_{u_a} - (\mathbf{w}_\Sigma)_{u_{a^*}}. \quad (3)$$

For simplicity, we assume that the original monomer set M does not contain any unit monomer u_a or u_{a^*} so that $|M'| = |M| + |\Sigma|$, which significantly simplifies the details of this section. However, our results continue to hold without this assumption with minor modification of the argument.²

A polymer $\mathbf{x} \in \mathbb{Z}^M$ can be *neutralized* by attaching the unit monomers to unbound domains. More precisely, we define the neutralizing map $\nu : \mathbb{Z}^M \rightarrow \mathbb{Z}^{M'}$ as follows.

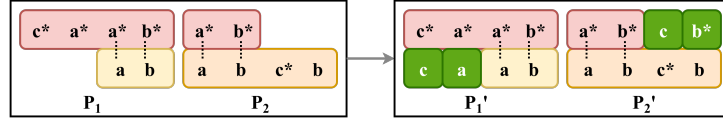
$$(\nu(\mathbf{x}))_m = \begin{cases} \mathbf{x}_m & \text{for } m \in M \\ \max(-(A\mathbf{x})_a, 0) & \text{for } m = u_a \\ \max((A\mathbf{x})_a, 0) & \text{for } m = u_{a^*} \end{cases}$$

In words, for each pair (a, a^*) , we add c copies of u_{a^*} if $\mathbf{x}$ has $c > 0$ unbound domain a , and add c copies of u_a if $\mathbf{x}$ has $c > 0$ unbound a^* . Figure 2 visualizes the neutralizing map.

We give the algebraic interpretation. For each $a \in D$, let $e_a \in \mathbb{Z}^D$ denote the unit vector with a 1 in the coordinate corresponding to a and 0 in all other coordinates. Observe that the neutralizing map indeed makes the net domain vector zero:

$$A' \cdot \nu(\mathbf{x}) = A\mathbf{x} + \left(\sum_{a \in D} \max(-(A\mathbf{x})_a, 0) \cdot e_a \right) + \left(\sum_{a \in D} \max((A\mathbf{x})_a, 0) \cdot (-e_a) \right) = \mathbf{0}.$$

² For example, we can add “dummy” domains to the singleton monomers in the original system to enforce the assumption.



■ **Figure 2** The polymers in the first configuration have been neutralized by unit monomers so that each vector representation has zero net domain vector.

This follows from Equation (3) coordinate-wise for each $a \in D$. This means that all the domains are bound, justifying the name of the neutralizing map ν . In general, we define the set of *neutral* polymers³ in the augmented system:

$$\mathcal{C}(A') = \{ \mathbf{z} \in \mathbb{N}^{M'} \mid A' \mathbf{z} = \mathbf{0} \}.$$

Each $\mathbf{z} \in \mathcal{C}(A')$ has a zero net domain vector. The set of neutral polymers has a clean algebraic representation, allowing us to relate the Pareto-optimal polymers in the system with the “indecomposable” elements, i.e., the Hilbert basis elements as we will see later.

Given a neutral polymer in the augmented system, we can recover a polymer in the original system using a projection map $\pi : \mathbb{Z}^{M'} = \mathbb{Z}^M \times \mathbb{Z}^\Sigma \rightarrow \mathbb{Z}^M$ that discards the unit monomer coordinates. In particular, $\pi(\nu(\mathbf{x})) = \mathbf{x}$ holds for all polymers $\mathbf{x} \in \mathbb{Z}^M$.

It is clear that $(\nu(\mathbf{x}))_{u_a} \cdot (\nu(\mathbf{x}))_{u_{a^*}} = 0$ for all $a \in D$. We call a polymer $\mathbf{w} \in \mathbb{Z}^{M'}$ *dichotomous* if it satisfies this property: $(\mathbf{w})_{u_a} \cdot (\mathbf{w})_{u_{a^*}} = 0$ for all $a \in D$. The following lemma is not hard to show. Intuitively, among all neutral augmented polymers that project to a fixed original polymer $\mathbf{x}$, there is exactly one with no redundant canceling pair of unit monomers.

► **Lemma 7.** $\nu(\mathbf{x})$ is the unique dichotomous neutral polymer in $\pi^{-1}(\mathbf{x}) \cap \mathcal{C}(A')$.

4.1 Hilbert Basis

We briefly recall the definitions and basic properties of Hilbert bases that we need. First, for an integer matrix $B \in \mathbb{Z}^{m \times n}$, the set $C = \{ \mathbf{x} \in \mathbb{R}^n : B\mathbf{x} \geq \mathbf{0} \}$ defined by a system of inequalities is called a (rational polyhedral) cone. If there exists a $\mathbf{c} \in \mathbb{R}^n$ such that $\mathbf{c} \cdot \mathbf{x} > 0$ for all $\mathbf{x} \in C \setminus \{ \mathbf{0} \}$, we say C is a pointed cone.

The following definition of the Hilbert basis will be convenient for this paper.

► **Definition 8** (Hilbert basis). The Hilbert basis $H(C)$ of a pointed cone C is a subset of $C \cap \mathbb{Z}^n \setminus \{ \mathbf{0} \}$ defined as follows:

$$H(C) = \{ \mathbf{h} \in C \cap \mathbb{Z}^n \setminus \{ \mathbf{0} \} \mid \mathbf{h} \text{ is not the sum of two vectors in } C \cap \mathbb{Z}^n \setminus \{ \mathbf{0} \} \}.$$

It is known that the Hilbert basis of any pointed (rational polyhedral) cone is finite. There are several equivalent definitions of the Hilbert basis. Notably, it is the smallest set that generates $C \cap \mathbb{Z}^n$ by nonnegative integer combinations; this characterization was previously used to enumerate the minimal set of reactions or polymers [2, 15].⁴ The computation of the Hilbert basis has been extensively studied in the literature [6, 16, 18], and implementations are publicly available [1, 5].

³ In other words, the saturated polymers. We stress that we do not assume that all polymers are saturated; we consider this set only in the augmented system for deriving the computational tools.

⁴ Technically, our definition coincides with this usual definition only for the *pointed* cone.

4.2 Pareto-Optimal Polymers as Hilbert Basis

The set of neutral polymers $\mathcal{C}(A')$ is an intersection of $\mathbb{Z}^{M'}$ and a cone

$$\mathcal{C}_{\mathbb{R}}(A') = \{\mathbf{x} \in \mathbb{R}^{M'} \mid A'\mathbf{x} \geq \mathbf{0}, -A'\mathbf{x} \geq \mathbf{0}, I\mathbf{x} = \mathbf{x} \geq \mathbf{0}\} \quad (4)$$

which is pointed because $\mathbf{1} \cdot \mathbf{x} > 0$ holds for all nonzero $\mathbf{x} \in \mathcal{C}_{\mathbb{R}}(A')$. We denote $H = H(\mathcal{C}_{\mathbb{R}}(A'))$ the Hilbert basis of $\mathcal{C}_{\mathbb{R}}(A')$, which is a subset of $\mathcal{C}(A') = \mathcal{C}_{\mathbb{R}}(A') \cap \mathbb{Z}^{M'}$ by definition.

Recall P^* is the set of Pareto-optimal polymers in the original system (Σ, M) . The following theorem states that the Pareto-optimal polymers are characterized by the projection of Hilbert basis elements $\pi(H) := \{\pi(\mathbf{h}) \mid \mathbf{h} \in H\}$.

► **Theorem 9.** $P^* = \pi(H) \setminus \{\mathbf{0}\}$.

The proof is deferred to Appendix A.3. A few implications are in order. First, this shows that P^* is finite because of the finiteness of the Hilbert basis. Furthermore, we can, in principle, compute the set P^* using the well-established Hilbert basis algorithms. We discuss the limitations of using this theorem directly in Appendix B.

5 Scalable Enumeration with Bounded Support

We present scalable algorithms to compute a meaningful subset of Pareto-optimal polymers. To scale the enumeration of Pareto-optimal polymers, we consider the subset of polymers with a bounded number of monomer types. We define the *support* of polymer $\mathbf{p} \in \mathbb{N}^M$ by

$$\text{supp}(\mathbf{p}) = \{m \in M \mid \mathbf{p}_m > 0\},$$

and for a user-chosen parameter t , the *t-bounded Pareto-optimal set* by

$$P_t^* = \{\mathbf{x} \in P^* \mid |\text{supp}(\mathbf{x})| \leq t\}.$$

We also consider a similarly restricted subset of P^* based on the number of domain pairs instead of monomers. This introduces the domain-support mode, which proved ineffective in our numerical experiments. One reason is that many polymers have a larger number of domain pairs than distinct monomers. In other words, the domain mode may be advantageous when the number of domain pairs is smaller than the number of monomers, or when many monomers share the same domain pairs. We outline this approach in Appendix C.

In Section 5.1, we show how to compute the t -bounded Pareto-optimal set through the computations of smaller Hilbert bases. However, the number of Hilbert bases to compute is large, making the naive algorithm less appealing. Instead, we introduce our main algorithm in Section 5.2, which optimizes the overall time complexity by reducing the number of Hilbert basis computations using covering designs. In numerical experiments of Section 7, we will see that the equilibrium concentration analysis with the subset P_t^* is almost as good as the one with the full set P^* for many applications.

5.1 Enumeration with All t -sized Supports

We introduce a few pieces of mathematical notation that will streamline the arguments to follow. Let $S \subseteq M$ be a subset of monomers with $|S| = t$. The set (Σ, S) gives a natural domain-monomer (sub)system. Define an *embedding* $\iota_S : \mathbb{N}^S \rightarrow \mathbb{N}^M$ that appends 0 on each entry corresponding to $M \setminus S$. After embedding, the set of polymers becomes

$$\text{Supp}(S) = \{\mathbf{p} \in \mathbb{N}^M : \text{supp}(\mathbf{p}) \subseteq S\}.$$

12:10 Scalable Enumeration of Pareto-Optimal Polymers

It is not hard to see that if $\mathbf{p} \in \text{Supp}(S)$ admits a split $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ with nonzero $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{N}^M$, then $\mathbf{p}_1, \mathbf{p}_2 \in \text{Supp}(S)$ as well, since the non-negativity of entries in $\mathbf{p}$ forces $\text{supp}(\mathbf{p}_1), \text{supp}(\mathbf{p}_2) \subseteq \text{supp}(\mathbf{p}) \subseteq S$.

Define $P_S^* = P^* \cap \text{Supp}(S)$ as the set of Pareto-optimal polymers whose support is contained in S . The following shows the connection between the Pareto-optimal polymers in the full system (Σ, M) and subsystem (Σ, S) . Intuitively, Pareto-optimality of the polymers supported only on S can be checked entirely inside the subsystem (Σ, S) , as any split of such a polymer is automatically also supported on S . We defer the formal proof to Appendix A.4.

► **Lemma 10.** *P_S^* is the set of Pareto-optimal polymers of the system (Σ, S) after embedding.*

In particular, this lemma implies the following inclusion:

$$P_S^* \subseteq P_T^* \text{ if } S \subseteq T. \quad (5)$$

Given this, we compute a subset of Pareto-optimal polymers, namely P_S^* , by computing a much smaller Hilbert basis. Formally, we define the set of neutral polymers restricted to S by

$$c^S(A') = \{ \mathbf{x} \in \mathbb{N}^{M'} \mid A' \mathbf{x} = \mathbf{0}, \text{supp}(\pi(\mathbf{x})) \subseteq S \},$$

and the corresponding pointed cone by

$$c_{\mathbb{R}}^S(A') = \{ \mathbf{x} \in \mathbb{R}^{M'} \mid A' \mathbf{x} = \mathbf{0}, \mathbf{x} \geq \mathbf{0}, \mathbf{x}_m = 0 \text{ for } m \in M \setminus S \} \quad (6)$$

following Equation (4). We denote $H_S = H(c_{\mathbb{R}}^S(A'))$ the Hilbert basis of $c_{\mathbb{R}}^S(A')$. Applying Theorem 9 on the system (Σ, S) , we have the following Hilbert basis characterization.

► **Lemma 11.** *$P_S^* = \pi(H_S) \setminus \{\mathbf{0}\}$. In particular, $P_t^* = \bigcup_{\substack{S \subseteq M \\ |S|=t}} (\pi(H_S) \setminus \{\mathbf{0}\})$.*

The proof is deferred to Appendix A.4. This lemma says that each support-restricted Pareto-optimal set can be computed by applying the Hilbert-basis characterization to the subsystem on S , and the t -bounded set is obtained by taking the union over these.

For $|S| \ll |M|$, computing a single Hilbert basis H_S can be done much more efficiently than computing the full Hilbert basis H . The effective ambient space of the pointed cone in Equation (6) is much smaller than that of the original cone in Equation (4), since we can ignore the always-zero original-monomer coordinates for $m \in M \setminus S$. More precisely, the dimensions are $|S| + |\Sigma|$ and $|M| + |\Sigma|$, respectively. Some domains may not appear in the monomers in S , so the dimension can be further reduced. Let D_S be the set of domains that appear in the monomers in S . Let $M'_S = S \cup \{u_a, u_{a^*} : a \in D_S\}$. We also reduce the number of equations defining the cone by considering $A'_S \in \mathbb{Z}^{D_S \times M'_S}$, obtained by removing irrelevant rows and columns. This significantly reduces the number of input constraints in computing H_S .

By Lemma 11, we can compute the t -bounded Pareto-optimal set P_t^* by multiple smaller Hilbert bases H_S for $|S| = t$. This naive subset enumeration algorithm requires one invocation of Hilbert basis computation per t -subset of M , giving a total of $\binom{n}{t}$ invocations. The factor $\binom{n}{t}$ grows rapidly with n and t , making naive enumeration infeasible for moderate n and t . In the next subsection, we explore a better strategy for selecting which subsets to run Hilbert basis computation on in order to reduce the total time.

5.2 Enumeration with Covering Design

The starting point of this section is Equation (5). This inclusion implies that if we compute a Hilbert basis H_T for some T , then we learn *all* H_S for all $S \subset T$. Instead of computing all H_S for $|S| = t$, we compute a small number of H_T with $|T| = k$ for some $k > t$. To minimize the complexity, we want to minimize the number of different T 's.

The *minimum covering design problem* addresses this problem. This problem asks for a collection $\mathcal{B}$ of k -element subsets (blocks) of M of minimum cardinality such that every t -element subset of M is contained in at least one block. More formally:

► **Definition 12** (Covering design). *Let M be a finite set of size n . A (n, k, t) -covering design $\mathcal{D}$ is a collection of k -element subsets of M such that for every t -element subset $T \subseteq M$, there exists a block $B \in \mathcal{D}$ with $T \subseteq B$. The covering number $c(n, k, t)$ denotes the minimum cardinality of any such collection.*

Let $\mathcal{D}$ be a $(|M|, k, t)$ -covering design. Given $\mathcal{D}$, the *covering design strategy* runs the Hilbert basis algorithm on each block $B \in \mathcal{D}$ and computes

$$\hat{P}_{k,t} = \bigcup_{B \in \mathcal{D}} (\pi(H_B) \setminus \{\mathbf{0}\}).$$

The following theorem shows that the covering design strategy computes (a superset of) the t -bounded Pareto-optimal set. The proof can be found in Appendix A.4.

► **Theorem 13.** $P_t^* \subseteq \hat{P}_{k,t} \subseteq P^*$.

In the covering-design strategy, using a covering design $\mathcal{D}$ requires $|\mathcal{D}|$ Hilbert basis computations. The Schönheim lower bound [19] gives

$$c(n, k, t) \geq \left\lceil \frac{n}{k} \left\lceil \frac{n-1}{k-1} \cdots \left\lceil \frac{n-t+1}{k-t+1} \right\rceil \cdots \right\rceil \right\rceil.$$

In practice, covering designs are vastly smaller than the family of all k -subsets, whose cardinality is $\binom{n}{k}$. This reduction makes the covering-design strategy substantially more attractive. However, computing the minimal covering design or computing $c(n, k, t)$ is known to be hard [9]. We use precomputed near-optimal covering designs, or greedy constructions if these are unavailable. Further details are described in the next section.

6 Implementation

All code is available at github.com/architrahul/Pareto-polymer-enumerator. We leverage the modern Hilbert basis algorithm `Normaliz` [5].

For a fixed size of the monomer set n and Pareto-optimal polymer support-bound t , the total runtime of the covering design strategy for a given k is determined by the number of `Normaliz` invocations $|\mathcal{D}_{n,k,t}|$ and the per-invocation cost, which depends on the specific sub-system. The user may specify a particular value of k directly, or use a built-in adaptive probing heuristic to select it. The probing heuristic exploits the fact that the average runtime of a `Normaliz` call for a fixed block size k can be estimated from a relatively small number of sampled blocks. Starting with small k , our tool runs `Normaliz` on a sample of blocks and uses the resulting average per-block runtime to extrapolate the total runtime, stopping when the runtime appears to be the minimum. Since the probing step uses only a small number of sampled blocks, its runtime is negligible compared with the full enumeration time of the experiment.

The details of the construction of covering designs and the block-size selection heuristic are described in Section 1 of the supplementary material.

7 Benchmarking

In this section, we present benchmarks of our new algorithm by applying the covering design strategy on several families of domain-monomer systems drawn from the thermodynamic binding network and DNA molecular programming literature [22, 23, 21]. Our analysis shows that the approach described in Section 6 can analyze large systems that could not be handled by previous TBN analysis methods or by the naive approach from Theorem 9.

We design the experiments to address two key aspects of the covering design approach, namely *scalability* and *equilibrium accuracy*. Our experiments show that, for many applications, the covering design strategy with an appropriate value of t and our heuristic choice of k quickly predicts the equilibrium concentration of the system. For example, for the AND gate linear cascade system [22] with 7 modules, the covering design approach with $(t, k) = (5, 25)$ computes the set $\hat{P}_{25,5}$ in 23.79s while the full Pareto-optimal set is computed in 1105.05s (Figure 3), achieving a $46\times$ speedup. The concentration analysis with COFFEE shows that the covering design approach does not miss any relevant polymer, as shown in Figure 5b.

In our experiments, all Hilbert basis computations were performed using `Normaliz` 3.11.1. The Hilbert basis computations dominate the overall time complexity; the other steps, including the heuristic choice of k , took relatively little time. Equilibrium concentrations were computed with COFFEE [24]. For the AND gate linear cascade experiments, we use two concentration regimes adapted from Figure S6 of the supplementary materials of Wang et al. [22]. In both regimes, the upstream inputs $X_2, \dots, X_{m+1}$ are supplied at 10 nM. Regime A sets all non-input monomers to 100 nM. Regime B uses the same concentrations as Regime A, but additionally supplies 50 nM of all monomers except the output of the final module. For each regime, we consider both a correct-output condition, where X_1 is supplied at 10 nM, and a leakage condition, where $X_1 = 0$.⁵ Each complementary domain bond is assigned energy either -20 kcal/mol or -10 kcal/mol.⁶ The value -20 kcal/mol represents a strong-binding regime comparable to the saturated regime used by Wang et al. [22], while -10 kcal/mol gives a weaker-binding system in which unsaturated polymers are more pronounced.⁷ All experiments were conducted on a MacBook Pro (Apple M3 Max, 16 cores, 48 GB RAM).

7.1 TBN Families

To perform numerical experiments and benchmarking, we focused on several families of TBNs. By “family,” we mean a collection of related TBNs in which the same motif is repeated arbitrarily many times. This parametrization lets us scale these systems to arbitrarily large sizes, increasing the difficulty of the problem.

Linear cascade of AND-gate modules

The AND-gate module was introduced in prior works [11, 4] and recently realized experimentally as a reversible signal propagation module [22]. The linear cascade system [22] (refer to Figure S3 of its supplementary materials for a diagram) consists of m sequentially

⁵ The theoretical model uses dimensionless mole fractions, but the COFFEE input concentrations are specified in molar units. In the dilute-solution regime considered here, mole fractions and molarities differ by a (temperature-dependent) constant conversion factor. COFFEE converts between molarity and mole fractions internally.

⁶ The total energy of a polymer is computed by summing this bond-energy contribution over all complementary domain bonds present in the polymer.

⁷ Although we used uniform bond energies across all domains, our framework does not require this; different domains can have different bond energies.

chained AND-gate modules. Each module is a collection of monomers that are normally bonded with each other in a particular way and, in the presence of two input polymers, can rearrange to release an output polymer. The output of each module serves as one of the two inputs to the next. Ideally, a module’s output forms only when both of its inputs are present, implementing an AND-gate computation. However, unintended reaction pathways cause outputs to form even in the absence of the required inputs, or produce other spurious polymers, a phenomenon known as *leakage*. This system is referred to as the AND gate linear cascade system in this paper.

Binary tree of AND-gate modules

The binary tree system [22] arranges AND-gate modules in a complete binary tree of depth d . Each module operates identically to those in the linear cascade, but here both inputs to every non-leaf module are outputs of two child modules, rather than a mix of primary inputs and upstream outputs. The number of monomers grows exponentially in d . This system is referred to as the AND gate binary tree system in this paper.

Scaffolded DNA system

Sterin et al. [21] introduced “scaffolded DNA computation,” which obtains the desired output at thermodynamic equilibrium. This system differs from the AND module motif in that large polymers are expected: the system consists of a large “scaffold” with short monomers bound to it, their number proportional to the length of the scaffold.

2-reactant, 2-product reversible signal propagation module (2-2 modules) system

This module from [23] behaves similarly to the AND-gate module in the linear cascade, with the difference that each module takes two input monomers and produces two output monomers. For the benchmarking system, both outputs of a module are used as inputs to the next module in the cascade. It will be referred to as the 2-2 linear cascade in this paper.

7.2 Runtime Analysis

Figures 3 and 4 show the (logarithmic) runtime of our algorithm for the AND gate linear cascade, binary tree system, and 2-2 linear cascade. We vary the system sizes by scaling the depth or the number of modules. In each system, we run the algorithm for various t , with the best k chosen by probing (described in section 1 of supplementary text).

Across all benchmark families, the covering design strategy provides substantial speedups of several orders of magnitude over full-system Hilbert basis computation (FULL HB), which is executed whenever feasible. The speedup is more drastic for larger systems, while for smaller systems, the covering design strategy with large t shows slower performance. The Hilbert basis computation dominates the overall runtime for all experiments.

7.3 Equilibrium Analysis Accuracy

We compute the equilibrium concentrations using COFFEE [24], given the set of polymers computed using either the covering design approach ($\hat{P}_{k,t}$) or the full Pareto-optimal set (P^*). For the systems considered in this section, the numerical experiments below show that $t = 5$ recovers almost all relevant polymers in equilibrium.

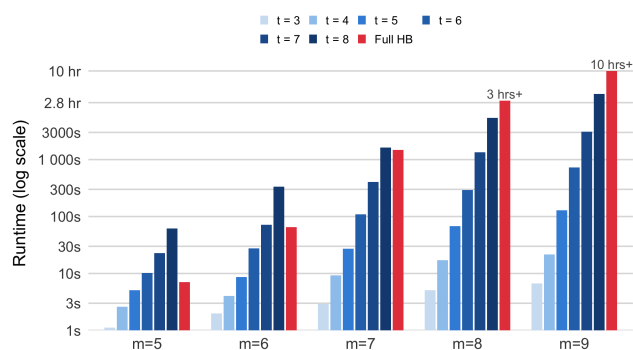


Figure 3 Pareto-optimal polymer set computation time for AND gate linear cascade systems with $m = 5-9$ modules. Within each group, bars indicate covering design $t = 3-8$ and full Hilbert basis computation (FULL HB). Bars labeled *hrs+* denote terminated runs, truncated at the cutoff.

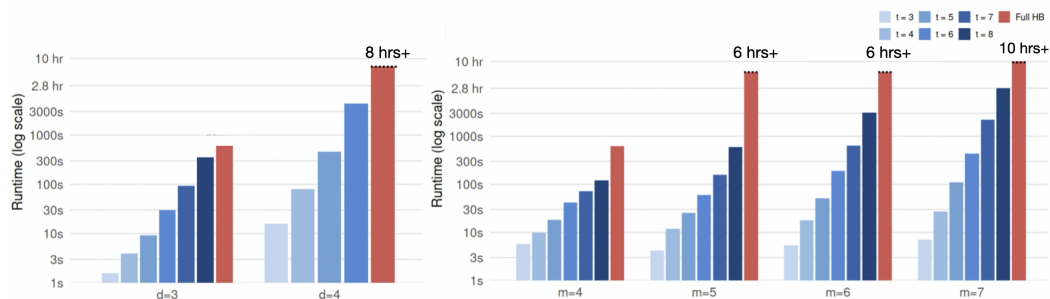


Figure 4 Pareto-optimal polymer set computation time for AND gate binary tree (left) and 2-2 linear cascade (right) TBN systems. Binary tree: depth $d = 3, 4$, with $t = 7, 8$ and full HB truncated for $d = 4$. 2-2 linear cascade system: $m = 4-7$, full HB truncated at 6 hours for $m = 5, 6$ and 10 hours for $m = 7$.

We focus on cases where computing the full Pareto-optimal set P^* is feasible to allow comparison with results on $\hat{P}_{k,t}$. In particular, we consider the AND gate linear cascade with 7 modules, the AND gate binary tree with depth 3, and the 2-2 linear cascade with 4 modules. The COFFEE analysis with the reduced candidates $\hat{P}_{k,t}$ for $t = 4, \dots, 7$ recovered all significant⁸ polymers from the full set, showing that the support-bounded restriction does not distort the analysis for these systems.

We also consider a seven-module AND gate linear cascade under the Regime B leakage condition with bond energy $E = -20$ kcal/mol. Here, the effect of the support bound t becomes visible, as shown in Figure 5. For $t = 3$, the reduced candidate set misses some equilibrium-relevant polymers and produces significant relative concentration errors for some recovered polymers. In contrast, for $t = 5$, all equilibrium-relevant polymers are recovered, and the relative concentration errors are comparatively very low.⁹

⁸ Polymers were categorized as significant if their equilibrium concentration exceeded 0.1 nM and insignificant otherwise.

⁹ Note that even for $t = 3$, only the lower-concentration polymers are missed and the overall equilibrium analysis is still fairly accurate.

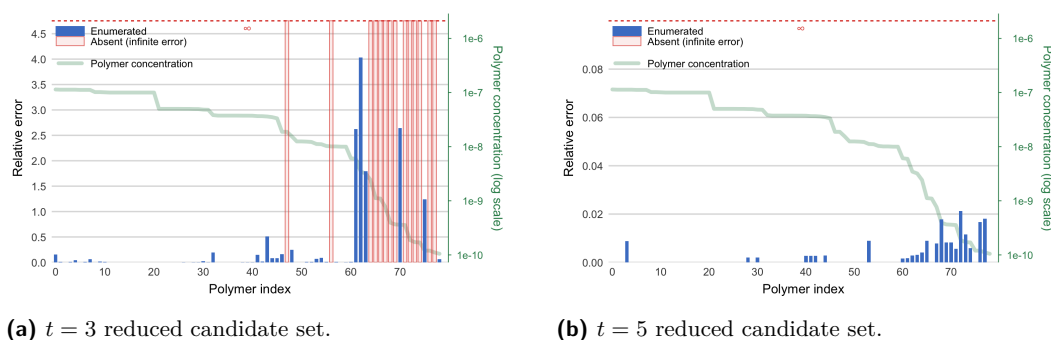


Figure 5 Relative concentration error for equilibrium-relevant polymers in the seven-module AND gate linear cascade under the Regime B leakage condition with bond energy $E = -20$ kcal/mol. Red lines indicate polymers absent from the reduced candidate set, corresponding to infinite relative error; blue lines indicate recovered polymers; the green line shows the polymer concentrations.

7.4 Leakage Analysis Applications

Beyond equilibrium analysis, the reduced candidate sets can be used for downstream applications as well. One important application is leakage analysis. In engineered TBN systems, leakage refers to the formation of an unintended product through spurious reaction pathways. We focus on the output leakage for the AND gate linear cascade family here, which is the final output concentration when any one of the inputs is removed.¹⁰ The output leakage analysis of this system with more modules is out of reach for the previous analytic tools.¹¹

We consider two leakage experiments. The first experiment compares how much leakage occurs depending on where the input is removed. We fix a seven-module AND gate linear cascade and remove one input at a time, using Regime B (introduced earlier in Section 7) with bond energy $E = -10$ kcal/mol. For each removed input, the equilibrium concentration of the final output polymer is computed using COFFEE on the candidate set produced by our covering-design enumeration pipeline. As shown in Section 2 of the supplementary materials, removing an input from the first module gives the largest leakage, and the leakage decreases as the removed input moves later in the cascade. This suggests that leakage generated earlier in the cascade can propagate through subsequent modules, whereas leakage introduced closer to the end has fewer downstream opportunities to amplify.

The second experiment studies how leakage changes as the cascade becomes longer. We remove an input to the first module and vary the number of modules in the linear cascade. For each setting, we compare the leakage output concentration against the correct output concentration obtained when all inputs are present.

We run this experiment under both concentration regimes (A and B) and both bond-energy values (-10 and -20 kcal/mol). Regime A gives the direct leakage setting, but the final output concentration reaches its plateau quickly. Regime B suppresses leakage and makes the dependence on cascade length more visible. Similarly, $E = -20$ kcal/mol mirrors the strong-binding regime used in Wang et al. [22] (the model used in [22], Section S4.2, restricts to saturated polymers only), while $E = -10$ kcal/mol tests a weaker-binding regime where unsaturated polymers play a larger role. Thus, the four cases in Figure 6 compare how leakage scales across both concentration and binding regimes.

¹⁰The mechanism of an AND gate linear-cascade module is illustrated in Figure 7; a full four-module example is given in Appendix D.1.

¹¹We note that the numerical analysis of [22] (Section S4.2) restricts to the fully bound (saturated) regime, and even with that simplification only extends up to 10 modules.

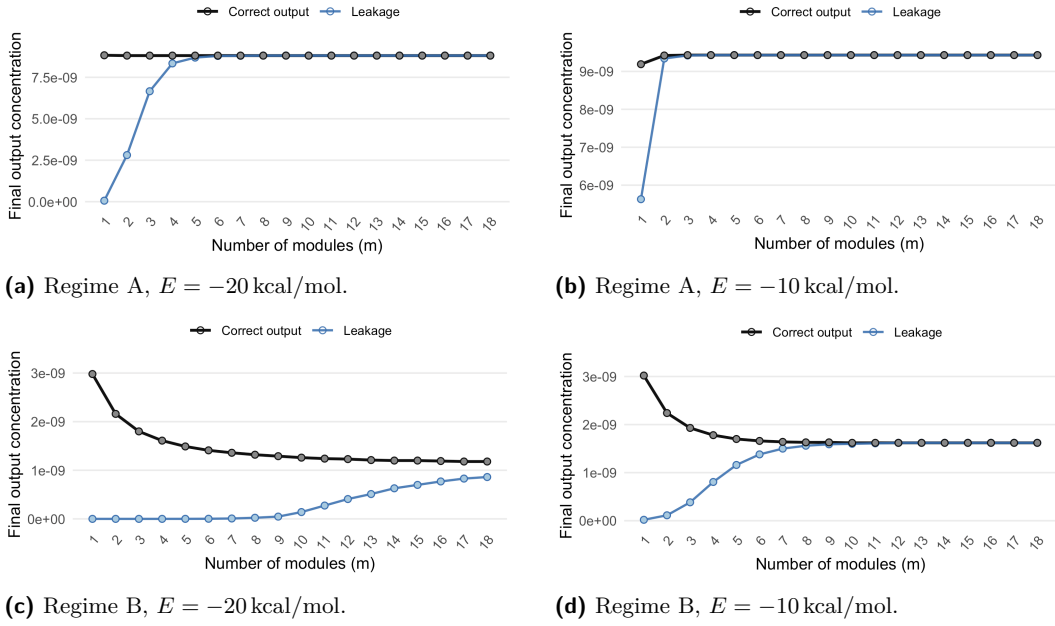


Figure 6 Leakage versus correct-output concentration for AND gate linear cascade systems across cascade length. The four panels show the two concentration regimes, A and B, crossed with bond energies $E = -20$ kcal/mol and $E = -10$ kcal/mol. For leakage, an input to the first module is removed; for correct output, all inputs are present. All curves are computed with COFFEE using $\hat{P}_{k,t}$, with $t = 5$ when feasible and $t = 4$ otherwise.

As shown in Figure 6, the final output concentration increases with cascade length. Regime A reaches a plateau quickly, whereas Regime B has comparatively less leakage. Weakening the bond energy from -20 kcal/mol to -10 kcal/mol increases leakage as expected.

Although the runtime still grows roughly exponentially with the system size, the covering-design approach remains practical for systems beyond the range of full Hilbert basis enumeration. When larger systems become intractable, the support parameter t can be decreased to further reduce the enumeration cost. This sacrifices some accuracy in the equilibrium analysis because fewer candidate polymers are included, but the earlier experiments show that small values of t can still recover the dominant equilibrium behavior in the tested systems. The main additional limitation is the size of the covering design: the dynamic-programming construction used for large parameters does not guarantee a minimum-size covering design, and may not scale optimally for very large $|M|$. Nevertheless, in our experiments, AND gate linear cascade systems with up to eighteen modules were analyzed efficiently enough to support leakage studies that would be infeasible with direct full Hilbert basis computation.

8 Conclusion

We introduced a framework for restricting polymer enumeration in geometry-free domain-monomer systems to the set of *Pareto-optimal polymers* for the purpose of equilibrium analysis. The Hilbert basis analysis provides a principled way to reduce the infinitely many possible polymers to finitely many that dominate the equilibrium. Further, because our method permits restricting the number of monomer types in allowed polymers, our algorithm can scale to larger systems for which enumerating all Pareto-optimal polymers would be impractical. Benchmarking our analysis on AND gate linear cascades, binary trees, and 2-2 linear cascade families from [22, 23, 21] shows the efficiency and accuracy of the algorithm for appropriate parameters.

The scalability of our algorithm is based on focusing on polymers with at most t distinct monomer types. This gives users a practical knob for trading completeness against runtime, tied to a design-level quantity rather than to an internal feature of the Hilbert-basis computation. Our experiments show that in many applications, modest values of t suffice to accurately predict the equilibrium behavior. However, when working with systems in which very large polymers are expected (as with the scaffolded DNA computer [21]), this support-bounded analysis is less effective.

Despite the greater efficiency afforded by our algorithm, our approach inevitably runs into limits for large t . One area for optimization is pre-computation of covering designs for larger system sizes. Recall that when a precomputed covering design is unavailable [13], we use the dynamic-programming construction of Gordon et al. [14]. As described in Section 1 of the supplementary material, this construction does not necessarily produce an optimal (smallest possible) covering design. Nonetheless, even with optimizations, the Hilbert basis algorithm becomes quickly impractical as the system size grows. The broader open question is whether, beyond the Pareto-optimality criterion introduced here, there are other structural properties – perhaps exploiting additional knowledge of a given system – that could further reduce the candidate set while preserving all equilibrium-relevant polymers.

References

- 1 4ti2 Team. 4ti2—a software package for algebraic, geometric and combinatorial problems on linear spaces. URL: <https://4ti2.github.io>.
- 2 Hamidreza Akef, Minki Hhan, and David Soloveichik. Computing and bounding equilibrium concentrations in athermic chemical systems. In *31st International Conference on DNA Computing and Molecular Programming (DNA 31)*, pages 10–1. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.DNA.31.10.
- 3 Keenan Breik, Cameron Chalk, David Doty, David Haley, and David Soloveichik. Programming substrate-independent kinetic barriers with thermodynamic binding networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(1):283–295, 2021. doi:10.1109/TCBB.2019.2959310.
- 4 Keenan Breik, Chris Thachuk, Marijn Heule, and David Soloveichik. Computing properties of stable configurations of thermodynamic binding networks. *Theoretical Computer Science*, 785:17–29, 2019. doi:10.1016/J.TCS.2018.10.027.
- 5 W. Bruns, B. Ichim, C. Söger, and U. von der Ohe. Normaliz: Algorithms for rational cones and affine monoids. <https://www.normaliz.uni-osnabrueck.de>.
- 6 Winfried Bruns and Bogdan Ichim. Normaliz: Algorithms for affine monoids and rational cones. *Journal of Algebra*, 324(5):1098–1113, 2010.
- 7 Kevin M Cherry and Lulu Qian. Supervised learning in DNA neural networks. *Nature*, 645(8081):639–647, 2025.
- 8 Evelyne Contejean and Hervé Devie. An efficient incremental algorithm for solving systems of linear diophantine equations. *Information and Computation*, 113(1):143–172, 1994. doi:10.1006/INCO.1994.1067.
- 9 Pilu Crescenzi, Federico Montecalvo, and Gianluca Rossi. Optimal covering designs: Complexity results and new bounds. *Discrete Applied Mathematics*, 144(3):281–290, 2004. doi:10.1016/J.DAM.2003.11.006.
- 10 Robert M Dirks, Justin S Bois, Joseph M Schaeffer, Erik Winfree, and Niles A Pierce. Thermodynamic analysis of interacting nucleic acid strands. *SIAM Review*, 49(1):65–88, 2007. doi:10.1137/060651100.
- 11 David Doty, Trent A Rogers, David Soloveichik, Chris Thachuk, and Damien Woods. Thermodynamic binding networks. In *International Conference on DNA-Based Computers*, pages 249–266. Springer, 2017. doi:10.1007/978-3-319-66799-7_16.

- 12 Mark E Fornace, Jining Huang, Cody T Newman, Avinash Nanjundiah, Nicholas J Porubsky, Marshall B Pierce, and Niles A Pierce. NUPACK: Analysis and design of nucleic acid structures, devices, and systems. *ChemRxiv*, 2025.
- 13 Daniel Gordon. La jolla coverings repository, April 2026. doi:10.5281/zenodo.19735294.
- 14 Daniel M Gordon, Oren Patashnik, and Greg Kuperberg. New constructions for covering designs. *Journal of Combinatorial Designs*, 3(4):269–284, 1995.
- 15 David Haley and David Doty. Computing properties of thermodynamic binding networks: An integer programming approach. In Matthew R. Lakin and Petr Šulc, editors, *27th International Conference on DNA Computing and Molecular Programming (DNA 27)*, volume 205 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:16, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DNA.27.2.
- 16 Raymond Hemmecke. On the computation of hilbert bases of cones. In *Mathematical Software: Proceedings of the First International Congress of Mathematical Software*, page 307. World Scientific, 2002.
- 17 Miki Hermann, Laurent Juban, and Phokion G Kolaitis. On the complexity of counting the hilbert basis of a linear diophantine system. In *International Conference on Logic for Programming Artificial Intelligence and Reasoning*, pages 13–32. Springer, 1999. doi:10.1007/3-540-48242-3_2.
- 18 Loic Pottier. The euclidean algorithm in dimension n. In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, pages 40–42, 1996. doi:10.1145/236869.236894.
- 19 Johanan Schonheim. On coverings. *Pacific Journal of Mathematics*, 14(4):1405–1411, 1964.
- 20 Tianqi Song and Lulu Qian. Heat-rechargeable computation in DNA logic circuits and neural networks. *Nature*, 646(8084):315–322, 2025.
- 21 Tristan Stérin, Abeer Eshra, Janet Adio, Constantine Glen Evans, and Damien Woods. A thermodynamically favoured molecular computer: Robust, fast, renewable, scalable. *bioRxiv*, 2025.
- 22 Boya Wang, Cameron Chalk, David Doty, and David Soloveichik. Molecular computation at equilibrium via programmable entropy. *Science Advances*, 12:eadx3969, 2026. doi:10.1126/sciadv.adx3969.
- 23 And Kaan Ata Yilmaz and David Soloveichik. Modularity of signal propagation networks in the thermodynamic binding network model. In *32nd International Conference on DNA Computing and Molecular Programming (DNA32)*, 2026.
- 24 Fu-Yao Yu, Sahil Shah, Yash Mittal, Paul Bessler, Aamir Mohsin, Jeffrey Geng, Arnav Vats, and David Soloveichik. COFFEE: A high-performance approach to convex optimization for thermodynamic equilibrium computations. In *2025 Systems and Information Engineering Design Symposium (SIEDS)*, pages 161–166. IEEE, 2025.

A

 Deferred Proofs

A.1 Proof of Theorem 6

Proof. Let $\mathbf{p} \in P \setminus P^*$ be a Pareto-suboptimal polymer. By definition, there exists a (nontrivial) split $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ with $\mathbf{p}_1$ and $\mathbf{p}_2$ not complementary. If either piece is itself suboptimal, split it again, continuing until all remaining polymers are Pareto-optimal. This process terminates because each split strictly decreases the number of monomers in each piece. Eventually this yields a decomposition of the form:

$$\mathbf{p} = \boldsymbol{\mu}_1 + \cdots + \boldsymbol{\mu}_k, \quad k \geq 2, \quad \boldsymbol{\mu}_i \in P^*,$$

where every split along the way is non-complementary. So we have the equilibrium relationship:¹²

$$[\mathbf{p}] = \prod_{i=1}^k [\mu_i]. \quad (7)$$

Fix one such decomposition for each $\mathbf{p} \in P \setminus P^*$ (although a suboptimal polymer may admit several decompositions). Note that, after fixing the decompositions, the map $\mathbf{p} \mapsto (\mu_1, \dots, \mu_k)$ is an injective map from the set of Pareto-suboptimal polymers to the set of tuples of Pareto-optimal polymers. Substituting (7) gives

$$C_{\text{sub}} = \sum_{\mathbf{p} \in P \setminus P^*} [\mathbf{p}] = \sum_{\mathbf{p} \in P \setminus P^*} \prod_{i=1}^{k(\mathbf{p})} [\mu_i(\mathbf{p})].$$

Each suboptimal polymer $\mathbf{p}$ contributes exactly one term to this sum, corresponding to its chosen tuple $(\mu_1, \dots, \mu_k)$ with $k \geq 2$. Since every such tuple is in particular an ordered tuple of elements of P^* , and all terms are non-negative, we may bound the sum by ranging over all ordered tuples of $k \geq 2$ elements of P^* .¹³

$$C_{\text{sub}} \leq \sum_{k=2}^{\infty} \sum_{\mu_1, \dots, \mu_k \in P^*} \prod_{i=1}^k [\mu_i].$$

The inner sum factorizes as a product of independent sums:

$$\sum_{\mu_1, \dots, \mu_k \in P^*} \prod_{i=1}^k [\mu_i] = \left(\sum_{\mu \in P^*} [\mu] \right)^k = C_{\text{opt}}^k.$$

Summing the resulting geometric series for $C_{\text{opt}} < 1$ completes the proof:

$$C_{\text{sub}} \leq \sum_{k=2}^{\infty} C_{\text{opt}}^k = \frac{C_{\text{opt}}^2}{1 - C_{\text{opt}}}. \quad \blacktriangleleft$$

A.2 Proof for Thermodynamic Justification

Proof of Lemma 5. Let C be a configuration containing a Pareto-suboptimal polymer $\mathbf{x}$. Then there exists a split $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$ with nonzero $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{N}^M$ such that $\mathbf{x}_1$ and $\mathbf{x}_2$ are not complementary. Let C' be the configuration obtained from C by replacing one copy of $\mathbf{x}$ with one copy each of $\mathbf{x}_1$ and $\mathbf{x}_2$. Since no bond is broken in this split, $H(C') = H(C)$. Since the number of separate polymer units increases by one, $S(C') = S(C) + 1$. Therefore

$$G(C') = a \cdot H(C') - b \cdot S(C') = G(C) - b < G(C),$$

so C' has strictly lower free energy than C . Since this holds for any configuration containing $\mathbf{x}$, no such configuration can be a minimum free energy configuration. $\blacktriangleleft$

¹²For example, consider the split $\mathbf{p} = \mu_1 + \mathbf{p}_2 = \mu_1 + (\mu_2 + \mu_3)$ where μ_1, μ_2, μ_3 are Pareto-optimal. The two intermediate splits $\mathbf{p} = \mu_1 + \mathbf{p}_2$ and $\mathbf{p}_2 = \mu_2 + \mu_3$ are non-complementary, so we have $[\mathbf{p}_2] = [\mu_2] \cdot [\mu_3]$ and $[\mathbf{p}] = [\mu_1] \cdot [\mathbf{p}_2]$ by Equation (2), which gives $[\mathbf{p}] = [\mu_1][\mu_2][\mu_3]$ as desired.

¹³For example, suppose $P^* = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and the suboptimal polymers are $\mathbf{a} + \mathbf{b}$, $\mathbf{a} + \mathbf{c}$, and $\mathbf{a} + \mathbf{b} + \mathbf{c}$ for simplicity. Then $C_{\text{sub}} = [\mathbf{a}][\mathbf{b}] + [\mathbf{a}][\mathbf{c}] + [\mathbf{a}][\mathbf{b}][\mathbf{c}]$, while the bound sums $([\mathbf{a}] + [\mathbf{b}] + [\mathbf{c}])^k$ over $k \geq 2$. The $k = 2$ term alone already exceeds the first two contributions, and the $k = 3$ term covers the third. Each power includes spurious terms (e.g. $[\mathbf{a}]^2$ or $[\mathbf{b}]^3$) that do not correspond to any suboptimal polymer, so the bound is loose, but all extra terms are non-negative.

A.3 Proof of Theorem 9

We observe the following lemma first.

► **Lemma 14.** *If $\mathbf{x} = \pi(\mathbf{h}) \in \pi(H) \setminus \{\mathbf{0}\}$, then $\mathbf{h} = \nu(\mathbf{x})$.*

Proof. Choose $\mathbf{h} \in H$ such that $\mathbf{x} = \pi(\mathbf{h})$. Observe that $\mathbf{e}_{u_a} + \mathbf{e}_{u_{a^*}} \in \mathcal{C}(A')$ for all $a \in D$. We will show that $\mathbf{h}$ is dichotomous, so it is the unique dichotomous neutral lift $\nu(\mathbf{x})$. Suppose that $(\mathbf{h})_{u_a} \cdot (\mathbf{h})_{u_{a^*}} \neq 0$ holds for some $a \in D$. Then both of them are positive integers because of $\mathbf{h} \geq \mathbf{0}$, which implies that $\mathbf{h} - \mathbf{e}_{u_a} - \mathbf{e}_{u_{a^*}} \in \mathcal{C}(A')$. This means that

$$\mathbf{h} = (\mathbf{e}_{u_a} + \mathbf{e}_{u_{a^*}}) + (\mathbf{h} - \mathbf{e}_{u_a} - \mathbf{e}_{u_{a^*}})$$

holds, which contradicts the definition of the Hilbert basis. This shows that $\mathbf{h}$ must be dichotomous. By the uniqueness in Lemma 7, $\mathbf{h}$ must be $\nu(\mathbf{x})$, so $\nu(\mathbf{x}) = \mathbf{h} \in H$. ◀

Proof of Theorem 9. We prove the two inclusions separately.

First inclusion $\pi(H) \setminus \{\mathbf{0}\} \subseteq P^*$. First, choose $\mathbf{x} \in \pi(H) \setminus \{\mathbf{0}\}$. Toward contradiction, assume that $\mathbf{x} \notin P^*$. This means that $\mathbf{x}$ is Pareto-suboptimal, so that there exists a non-complementary split $\mathbf{x} = \mathbf{y} + \mathbf{z}$ with nonzero $\mathbf{y}, \mathbf{z}$ such that $(A\mathbf{y})_a \cdot (A\mathbf{z})_a \geq 0$ for all $a \in D$. In other words, for each a , $(A\mathbf{y})_a, (A\mathbf{z})_a$, as well as $(A\mathbf{x})_a$ are all simultaneously nonnegative (i.e., ≥ 0) or simultaneously nonpositive (≤ 0).

Because all have the same sign, the neutralization $\nu(\mathbf{x}), \nu(\mathbf{y}), \nu(\mathbf{z})$ all add only one of u_a or u_{a^*} for each a . From this, we have $\nu(\mathbf{x}) = \nu(\mathbf{y}) + \nu(\mathbf{z})$. However, this is a contradiction because Lemma 14 shows that $\nu(\mathbf{x}) \in H$, which must not be written as the sum of two nonzero $\nu(\mathbf{y}), \nu(\mathbf{z}) \in \mathcal{C}(A')$. This shows that $\pi(H) \setminus \{\mathbf{0}\} \subseteq P^*$.

Second inclusion $P^* \subseteq \pi(H) \setminus \{\mathbf{0}\}$. Next, choose $\mathbf{x} \in P^*$, which is nonzero by definition, and assume that $\mathbf{x} \notin \pi(H) \setminus \{\mathbf{0}\}$. This shows that $\nu(\mathbf{x}) \notin H$, otherwise $\mathbf{x} = \pi(\nu(\mathbf{x})) \in \pi(H) \setminus \{\mathbf{0}\}$.

By the definition of the Hilbert basis, there are two nonzero $\mathbf{v}, \mathbf{w} \in \mathcal{C}(A')$ such that $\nu(\mathbf{x}) = \mathbf{v} + \mathbf{w}$, which gives a split $\mathbf{x} = \pi(\mathbf{v}) + \pi(\mathbf{w})$. We will show that this is a non-complementary split so that $\mathbf{x}$ is Pareto-suboptimal.

Given $\mathbf{v}, \mathbf{w} \geq \mathbf{0}$ and the dichotomous property of $\nu(\mathbf{x})$, the split polymers $\mathbf{v}, \mathbf{w}$ must also be dichotomous. Furthermore, for each $a \in D$, one of the two unit-monomer coordinates is zero in all three vectors:

$$(\nu(\mathbf{x}))_{u_a} = 0 \text{ and } (\mathbf{v})_{u_a} = (\mathbf{w})_{u_a} = 0, \text{ or } (\nu(\mathbf{x}))_{u_{a^*}} = 0 \text{ and } (\mathbf{v})_{u_{a^*}} = (\mathbf{w})_{u_{a^*}} = 0. \quad (8)$$

Suppose that $\pi(\mathbf{v}) = \mathbf{0}$ so that $\mathbf{x} = \pi(\mathbf{w})$. The polymer $\mathbf{w}$ is a dichotomous neutral element in $\pi^{-1}(\mathbf{x}) = \pi^{-1}(\pi(\mathbf{w}))$, which must be $\nu(\mathbf{x})$ by Lemma 7, leading to a contradiction: $\mathbf{v} = \nu(\mathbf{x}) - \mathbf{w} = \mathbf{0}$. Therefore $\pi(\mathbf{v}) \neq \mathbf{0}$ and similarly $\pi(\mathbf{w}) \neq \mathbf{0}$.

It remains to show that $\pi(\mathbf{v})$ and $\pi(\mathbf{w})$ are non-complementary. By Equation (3),

$$(A\pi(\mathbf{w}))_a + (\mathbf{w})_{u_a} - (\mathbf{w})_{u_{a^*}} = (A'\mathbf{w})_a = 0$$

holds for all $a \in D$, where the last equality is due to $\mathbf{w} \in \mathcal{C}(A')$. By Equation (8), one of the following must be true:

$$(A\pi(\mathbf{w}))_a + (\mathbf{w})_{u_a} = (A\pi(\mathbf{v}))_a + (\mathbf{v})_{u_a} = 0 \text{ or } (A\pi(\mathbf{w}))_a - (\mathbf{w})_{u_{a^*}} = (A\pi(\mathbf{v}))_a - (\mathbf{v})_{u_{a^*}} = 0.$$

In any case, $(A\pi(\mathbf{w}))_a$ and $(A\pi(\mathbf{v}))_a$ are both nonnegative or nonpositive because of $\mathbf{v}, \mathbf{w} \geq \vec{0}$, so that $(A\pi(\mathbf{w}))_a \cdot (A\pi(\mathbf{v}))_a \geq 0$ for all $a \in D$. This proves that $\pi(\mathbf{v})$ and $\pi(\mathbf{w})$ are non-complementary. Since $\pi(\mathbf{v}), \pi(\mathbf{w}) \neq \mathbf{0}$ and $\mathbf{x} = \pi(\mathbf{v}) + \pi(\mathbf{w})$, this is a non-complementary split of $\mathbf{x}$, contradicting $\mathbf{x} \in P^*$. Therefore $\mathbf{x} \in \pi(H) \setminus \{\mathbf{0}\}$.

Combining the two inclusions, we conclude the proof. $\blacktriangleleft$

A.4 Proof for Covering Design Strategy

Proof of Lemma 10. First, if $\mathbf{p} \in P_S^* \subset P^*$, then any split $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ for nonzero $\mathbf{p}_1, \mathbf{p}_2 \in \text{Supp}(S) \subset \mathbb{N}^M$ must be complementary. Therefore, $\mathbf{p}$ is a Pareto-optimal polymer in (Σ, S) after embedding.

On the other hand, if $\mathbf{p}$ is a Pareto-optimal polymer in (Σ, S) after embedding, it is clear that $\mathbf{p} \in \text{Supp}(S)$. Also, any split of it $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ for nonzero $\mathbf{p}_1, \mathbf{p}_2 \in \text{Supp}(S)$ must be complementary. We showed that all splits must be in this form, thus $\mathbf{p}$ is in P^* . This proves the equivalence. $\blacktriangleleft$

Proof of Lemma 11. For the first statement, observe that the entries in $M \setminus S$ are always 0 and can be removed. After removal, this is exactly Theorem 9 on the system (Σ, S) .

For the second statement, any $\mathbf{p} \in P_t^*$ has $|\text{supp}(\mathbf{p})| \leq t$, so $\text{supp}(\mathbf{p})$ is contained in some subset $S \subseteq M$ with $|S| = t$. Hence $\mathbf{p} \in P_S^* = \pi(H_S) \setminus \{\mathbf{0}\}$. Conversely, if $\mathbf{p} \in \pi(H_S) \setminus \{\mathbf{0}\}$ for some $S \subseteq M$ with $|S| = t$, then $\mathbf{p} \in P_S^* \subseteq P^*$ and $|\text{supp}(\mathbf{p})| \leq |S| = t$, so $\mathbf{p} \in P_t^*$. $\blacktriangleleft$

Proof of Theorem 13. The inclusion $\subseteq P^*$ holds because $\pi(H_B) \setminus \{\mathbf{0}\} = P_B^* = P^* \cap \text{Supp}(B) \subset P^*$.

For the first inclusion, it suffices to show that $P_S^* \subseteq \hat{P}_{k,t}$ for all $|S| = t$ because of Lemma 11. By the definition of the covering design and Equation (5), there exists $B \in \mathcal{D}$ such that $S \subseteq B$ so that $P_S^* \subseteq P_B^*$. This concludes the proof. $\blacktriangleleft$

B Scalability Limitations

Theorem 9 shows that the set of Pareto-optimal polymers P^* is finite and computable. We can employ a modern implementation, such as `Normaliz`, to compute the Hilbert basis H , which gives the set of Pareto-optimal polymers for a relatively small domain-monomer system. However, this approach quickly becomes impractical as the system grows. We give some explanation of this phenomenon in the next subsection.

A natural way to remedy this inefficiency is to terminate the algorithm early, expecting to obtain a part of the full Hilbert basis. Unfortunately, the state-of-the-art implementations of Hilbert basis algorithms usually enumerate a superset of the Hilbert basis first and then iteratively reduce it to the Hilbert basis. This algorithmic aspect makes it difficult to introduce a meaningful stopping criterion, as we are not aware of any other way to understand the intermediate superset except for the reduction to the Hilbert basis.

We have made several attempts to employ scalable algorithms, which turned out to be inefficient. We explain two notable approaches below.

► **Remark 15 (Alternative representation).** We can use the cone $\{\mathbf{x} \in \mathbb{N}^{|M|+|D|} \mid A'\mathbf{x} \geq \mathbf{0}\}$ and its Hilbert basis by only adding the complementary unit monomers $\{u_{a^*}\}_{a \in D}$. In this setting, the net domain vector has nonnegative entries instead of being a zero vector. This formulation and computation ultimately gives the same result as in this paper, but modern implementations such as `Normaliz` are more native to the cone $A'\mathbf{x} = \mathbf{0}$, as we present in the main body.

B.1 Hilbert Basis Hardness

The size of the Hilbert basis rapidly increases along several parameters, and counting the number of Hilbert basis elements is known to be NP-hard (strictly speaking, #P-hard [17]). In fact, the Hilbert basis of a pointed cone defined by a tiny matrix can be arbitrarily large: the Hilbert basis of the pointed cone $\mathcal{C}_{\mathbb{R}}(A_k) = \{\mathbf{x} \in \mathbb{R}^3 \mid A_k \mathbf{x} = 0, \mathbf{x} \geq 0\}$ for $A_k = \begin{bmatrix} 1 & 1 & -k \end{bmatrix}$ is

$$H_k = \{(i, k - i, 1) \mid i = 0, \dots, k\}.$$

In this example, the size of H_k increases exponentially for the bit-length of describing A_k , i.e., the input size.

B.2 Failed Approaches

This subsection introduces two slow approaches, highlighting that it is not clear how to obtain a practical, scalable algorithm by either (i) employing an older scalable Hilbert basis algorithm or (ii) directly exploiting properties of the Pareto-optimal polymers. This motivates the main approach in Section 5, employing both modern Hilbert basis algorithms and properties of the Pareto-optimal polymers.

The first approach is based on an old Hilbert basis algorithm [8]. This algorithm inspects candidate basis elements by increasing the target 1-norm (i.e., the sum of absolute values of entries). This is exceedingly slow, especially because of the enormous number of small-norm polymers. The algorithm reduces the search size using some geometric ideas, but our implementation failed to terminate even for small examples.

Another approach is to use the following nontrivial fact for Pareto-optimal polymers.

► **Lemma 16.** *For every Pareto-optimal polymer $\mathbf{p} \in P^*$ that is not a monomer, there exist nonzero $\mathbf{p}_1, \mathbf{p}_2 \in P^*$ such that $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$.*

We will not prove this lemma here because it is not relevant to the main body of this paper. Given this lemma, we can recursively enumerate P^* by repeatedly merging pairs $\mathbf{p}, \mathbf{q}$ of previously enumerated Pareto-optimal polymers and checking whether the resulting $\mathbf{p} + \mathbf{q}$ is Pareto-optimal. Our implementation (with some optimizations) shows that this approach is also much slower than the simple Hilbert basis algorithm using Theorem 9.

C Domain Support Mode

This section introduces an alternative way to make the computation of the Pareto-optimal polymers scalable, based on the domain supports instead of the monomer supports. In the *domain mode*, subset enumeration is applied over domain pairs rather than monomer types. This may be advantageous when the number of domains is relatively smaller than the number of monomers, or when many monomers share the same domain pairs, as in cascade and binary-tree TBNs, in which case the relevant covering design size $c(|D|, k, t)$ may be significantly smaller than $c(|M|, k, t)$. Further details are provided in Section 4 of the supplementary material.

D TBN System Examples

D.1 Linear Cascade 4 Module AND Gate System

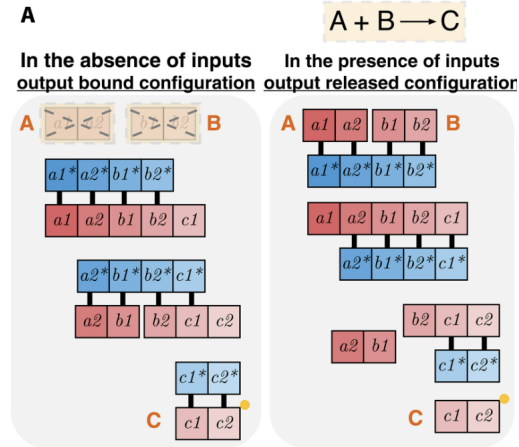


Figure 7 Mechanism of a single AND-gate module in the linear cascade system. In the presence of both input polymers, the module can rearrange to release the output polymer, which then serves as an input to the next module in the cascade. Figure adapted from Fig. 3 of [22].

```
inputs: x1_1 x1_2
        x2_1 x2_2
        x3_1 x3_2
        x4_1 x4_2
        x5_1 x5_2

# Module 1: x1 + x2 -> y1
x1_1 x1_2 x2_1 x2_2 y1_1
x1_1* x1_2* x2_1* x2_2*
x1_2 x2_1
x2_2 y1_1 y1_2
x1_2* x2_1* x2_2* y1_1*
y1_1* y1_2*
outputy1: y1_1 y1_2

# Module 2: y1 + x3 -> y2
y1_1 y1_2 x3_1 x3_2 y2_1
y1_1* y1_2* x3_1* x3_2*
y1_2 x3_1
x3_2 y2_1 y2_2
y1_2* x3_1* x3_2* y2_1*
y2_1* y2_2*
outputy2: y2_1 y2_2

# Module 3: y2 + x4 -> y3
y2_1 y2_2 x4_1 x4_2 y3_1
y2_1* y2_2* x4_1* x4_2*
y2_2 x4_1
x4_2 y3_1 y3_2
y2_2* x4_1* x4_2* y3_1*
y3_1* y3_2*
outputy3: y3_1 y3_2

# Module 4: y3 + x5 -> y4
y3_1 y3_2 x5_1 x5_2 y4_1
y3_1* y3_2* x5_1* x5_2*
y3_2 x5_1
x5_2 y4_1 y4_2
y3_2* x5_1* x5_2* y4_1*
y4_1* y4_2*
outputy4: y4_1 y4_2
```

D.2 2-2 Linear Cascade System

```

x11 x12 x13
x21 x22 x23
x11 x12 x13 x21 x22 x23 y11 z11
x11* x12* x13* x21* x22* x23*
x12 x13 x22 x23 y11 y12 z11 z12
x12* x13* x22* x23* y11* z11*
x13 x23 y11 y12 y13 z11 z12 z13
x13* x23* y11* y12* z11* z12*
y11* y12* y13* z11* z12* z13*
y11 y12 y13
z11 z12 z13
y11 y12 y13 z11 z12 z13 y21 z21
y11* y12* y13* z11* z12* z13*
y12 y13 z12 z13 y21 y22 z21 z22
y12* z13* z12* z13* y21* z21*
y13 z13 y21 y22 y23 z21 z22 z23
y13* z13* y21* y22* z21* z22*
y21* y22* y23* z21* z22* z23*
y21 y22 y23
z21 z22 z23

```

D.3 Scaffolded DNA System

```

s[1]* s[2]* s[3]* s[4]* s[5]* s[6]* s[7]* s[8]* s[9]* s[10]*
t0[1] s[1] t0[2]*
t1[1] s[1] t1[2]*
t0[2] s[2] t0[3]*
t1[2] s[2] t1[3]*
t0[3] s[3] t0[4]*
t1[3] s[3] t1[4]*
t0[4] s[4] t0[5]*
t1[4] s[4] t1[5]*
t0[5] s[5] t0[6]*
t1[5] s[5] t1[6]*
t0[6] s[6] t0[7]*
t1[6] s[6] t1[7]*
t0[7] s[7] t0[8]*
t1[7] s[7] t1[8]*
t0[8] s[8] t0[9]*
t1[8] s[8] t1[9]*
t0[9] s[9] t0[10]*
t1[9] s[9] t1[10]*
t0[10] s[10]
t1[10] s[10]

```

D.4 AND Gate Binary Tree System

```

input: x1_1 x1_2, x2_1 x2_2, x3_1 x3_2, x4_1 x4_2
# Module 1: x1 + x2 -> y1
x1_1 x1_2 x2_1 x2_2 y1_1
x1_1* x1_2* x2_1* x2_2*
x1_2 x2_1
x2_2 y1_1 y1_2
x1_2* x2_1* x2_2* y1_1*
y1_1* y1_2*
outputy1: y1_1 y1_2
# Module 2: x3 + x4 -> y2
x3_1 x3_2 x4_1 x4_2 y2_1
x3_1* x3_2* x4_1* x4_2*
x3_2 x4_1
x4_2 y2_1 y2_2
x3_2* x4_1* x4_2* y2_1*
y2_1* y2_2*
outputy2: y2_1 y2_2
# Module 3: y1 + y2 -> y3
y1_1 y1_2 y2_1 y2_2 y3_1
y1_1* y1_2* y2_1* y2_2*
y1_2 y2_1
y2_2 y3_1 y3_2
y1_2* y2_1* y2_2* y3_1*
y3_1* y3_2*
outputy3: y3_1 y3_2

```