

Bolzano: Case Studies in LLM-Assisted Mathematical Research

Jan Grebík¹ Pavel Hubáček^{1,2} Martin Koutecký¹ Matěj Kripner³
 Václav Rozhoň¹ Robert Šámal¹ Adrián Zámečník¹

¹Computer Science Institute, Charles University

²Institute of Mathematics, Czech Academy of Sciences

³Institute of Formal and Applied Linguistics, Charles University

Abstract

We report new results on six problems in mathematics and theoretical computer science, produced with the assistance of Bolzano—an open-source multi-agent LLM system. Bolzano orchestrates rounds of interaction between parallel prover agents and a verifier agent while maintaining a persistent knowledge base that is carried across rounds. Classified using the significance–autonomy taxonomy of Feng et al. [F⁺26b], four of the six results reach the level of publishable research, and three of the six were produced essentially autonomously by Bolzano. Our results provide evidence that LLMs can contribute meaningfully to mathematical research, complementing recent reports by Bubeck et al. [BCE⁺25], Woodruff et al. [WCAJ⁺26], and others.

1 Introduction

The integration of large language models into mathematical research has moved rapidly from speculation to documented practice. We highlight two recent reports that have established this as a real phenomenon (see Related Work for many more results in this direction):

- Woodruff, Mirrokni et al. [WCAJ⁺26] present a collection of case studies in which Google’s Gemini Deep Think solved open problems across mathematics, theoretical computer science, and physics.
- Bubeck et al. [BCE⁺25] document experiments in which OpenAI’s GPT-5 contributed to research in mathematics, physics, and computer science.

This paper provides more evidence in this direction. Bolzano [Bol25b] is an open-source AI system that assists with mathematical research and is available to interested researchers. The system runs research on a problem using several rounds of interaction between several provers and a verifier. Each prover and verifier is implemented by state-of-the-art LLMs (GPT, Gemini, Claude, or others). Its architecture is described in Section 2. We use Bolzano to provide new results for six problems in mathematics and theoretical computer science.

Strengths of Bolzano Across our case studies, Bolzano was most useful as a generator of mathematically meaningful intermediate research moves: it was particularly good at finding counterexamples and obstructions, proposing concrete constructions and gadgets, or extending a known base case or simpler proof template to a more general statement. This mirrors the findings in [WCAJ⁺26, BCE⁺25]. In one case, Bolzano autonomously provided a hardness proof and a polynomial-time algorithm for a restricted class, and corrected a small error in the human-provided problem formulation.

	H (Primarily human)	C (Collaboration)	A (Autonomous)
Negligible novelty			
Minor novelty			Partitioning (Section 3.5) Tilings (Section 3.2)
Publishable research	Heaps (Section 3.4)	PWPP (Section 3.1) KZG (Section 3.3)	KKOS Optimization (Section 3.6)
Major advance			
Landmark break-through			

Table 1: Classification of our results by significance and autonomy, following the taxonomy proposed by Feng et al. [F⁺26b]. to landmark breakthrough. Autonomy ranges from H (primarily human, secondary AI input) through C (human–AI collaboration) to A (essentially autonomous).

Limitations This work has several limitations. First, we do not provide a thorough comparison between Bolzano and single-session chatbot interactions; while such a comparison is feasible in principle, it is difficult to carry out rigorously in a case study with six problems, and the advantage of the multi-agent architecture over direct use of the same underlying models remains unquantified. Second, it is hard to cleanly disentangle the human and AI contributions; even selecting the right problem to work on is a nontrivial human input.

Solved problems We document six problems in mathematics and theoretical computer science that Bolzano has resolved or made progress on. This was done either autonomously or through collaboration with domain-expert mathematicians and computer scientists. For each of the problems, we include discussion. The corresponding formal results are proved either in the appendices of this paper or in separate papers.

1. **Complexity theory:** We construct an oracle separation showing that PWPP is not closed under adaptive Turing reductions (Section 3.1). The human input was the question, and after Bolzano returned a proof, a human researcher suggested to keep the main structure (the same choice of problem showing that PWPP is not closed) but try a different strategy for the analysis, after which Bolzano essentially finished the proof.
2. **Additive combinatorics:** We provide an example of a tiling of $\mathbb{R}^2$ by a single tile that shows a limit on the technique from a forthcoming paper [dDGM26] (Section 3.2).
3. **Cryptography:** We prove special soundness for multi-polynomial, multi-point KZG batching in the standard model (Section 3.3). The human researchers proposed a variant of the protocol and provided a simpler base-case proof as context. Bolzano found the full proof in six rounds, producing a lengthy but essentially complete formal argument.
4. **Data structures:** We prove that two beyond worst-case properties of heaps are equivalent (Section 3.4). This was first proven by a human researcher in a (yet unpublished) project. Bolzano independently reproved the equivalence and independently came up with a stronger statement and its proof.
5. **Combinatorics:** We disprove a conjecture on function-preimage partitioning from the KAMAK 2020 workshop [Krá20]. Moreover, Bolzano establishes a corrected hypothesis and proves some bounds for it (Section 3.5).

6. **Computational complexity:** We determine the complexity of an optimization problem over the KKOS cultural dynamics model [KKOS16] (Section 3.6). The decision version is NP-complete on general graphs; on forests, we give a polynomial-time algorithm. Bolzano worked autonomously, also correcting a small error in the original problem formulation.

In general, our experience parallels the findings of [WCAJ⁺26] and [BCE⁺25] in several respects. LLMs excel at generating proof candidates, constructing counterexamples, and making cross-domain connections. In early 2026, human guidance remains relevant for problem selection, high-level strategy, and verification.

Related work Recent work has begun to document the use of LLMs for mathematical and scientific research. GPT-5 has been shown to contribute to research in mathematics, physics, and other sciences, including solving open problems and improving constants [BCE⁺25]. Similar results have been reported with Gemini [WCAJ⁺26]. Aletheia [F⁺26b] is a mathematics research agent based on a Generator–Verifier–Reviser loop that autonomously solved research-level problems, including 6 out of 10 problems in the First Proof challenge [FJhK⁺26]. First Proof [ABH⁺26] is a benchmark of 10 research-level problems from unpublished work of 11 mathematicians; OpenAI [Ope26] reported at least 5 likely correct solutions. AlphaEvolve [N⁺25] has been applied to 67 problems [GGSTW25], mostly in constant optimization and construction search, with expert hints improving efficiency without clearly changing the ceiling of performance. FunSearch [RPBN⁺24] pairs an LLM with an evaluator in an evolutionary loop and has produced new results in combinatorics and algorithms.

Many systems pursue automated theorem proving with formal verification. AlphaGeometry [TWL⁺24] solves olympiad geometry problems via a neuro-symbolic engine. Many recent systems target the Lean proof assistant, including AlphaProof [HMS⁺25], which achieves IMO medal-level performance via reinforcement learning; Hilbert [V⁺25], which combines informal proof sketches with formal verification; Aleph Prover [Log25], a formal theorem prover that leads PutnamBench; Ax-Prover [B⁺25], which targets quantum physics and abstract algebra; and auto-formalization of a 130,000-line topology textbook at low cost [Urb26]. We view formal verification as complementary to our approach: Bolzano relies on informal proof generation followed by expert verification, which currently offers broader coverage but weaker guarantees. A large-scale evaluation of over 5,000 LLM-generated proofs across 1,000 problems [D⁺25] finds substantially stronger performance in informal than in formal proof generation.

Multi-agent architectures for mathematical reasoning have been explored in several directions, including multiagent debate, step-level verification, iterative self-refinement, and LLM scaffolding for problem solving [DLT⁺23, G⁺25, HY25, MTG⁺23]. In late 2025, open Erdős problems emerged as a testing ground for LLM mathematical capabilities, with models producing original proofs of previously unsolved problems [PSV26, F⁺26a]; however, the problems solved so far are generally amenable to straightforward techniques [F⁺26a].

Roadmap Section 2 describes the architecture and design of Bolzano. Section 3 presents each of the solved problems, including the problem statement, context, a discussion of the human–AI collaboration, and a link to the Bolzano transcript. For problems whose proofs are not contained in a separate paper, the full proofs appear in the appendices. All proofs have been verified by domain experts.

2 Bolzano

Bolzano is a research tool available at <https://bolzano.app>. It provides an automated pipeline consisting of *prover*, *verifier*, and *summarizer agents* designed to iteratively investigate mathematical research problems. Throughout this section, by *agent* we mean a callable function that makes a preconfigured request to an LLM with a custom prompt specifying a research persona, tasks, and goals.

Overall architecture We call one iteration of the Bolzano pipeline a *research round*. It involves running n parallel *prover agents*, followed by one *verifier agent*, and concluded by a *summarizer agent*. Research rounds run sequentially; between rounds, the state of the investigation is preserved in three human-readable files.

Agents. The *prover agents* are tasked with coming up with proof ideas, finding counterexamples, proving special cases, identifying mistakes, and writing proofs. The *verifier agent* checks the work of the provers—identifying errors and unjustified steps, combining ideas into viable solutions—and is the sole agent that decides what gets written into the knowledge base files. The *summarizer agent* produces a concise summary of each research round for the user and for subsequent agents.

Knowledge base files. In between research rounds, we maintain three files that serve as a persistent knowledge base. The *notes* file aggregates insights, failed approaches, conjectures, and simplified proofs. The *proofs* file maintains rigorous, fully detailed proofs. The *output* file provides a short summary of the current status for the human researcher. All agents read from these files, but only the verifier can write to them.

Model diversity. Bolzano allows the user to select a different LLM for each prover agent. Since models differ in training data, this seems to generate a wider variety of approaches. This approach is also reported to counteract self-preference bias [PBF24].

Human guidance. Between research rounds, the user may provide additional instructions to steer the investigation. In our experience, human guidance often leads to stronger results—for example, the PWPP result (Section 3.1) was obtained after an expert advised Bolzano to try a different strategy.

3 Solved Problems

This section contains the six selected problems solved by Bolzano. For each problem, we add discussion about what parts of the research pipeline have been done by a human, and which parts by Bolzano. We always add a link to the Bolzano system containing the proof. However, Bolzano-generated proofs are not intended to be publication-ready. We thus always either add expert-verified proof to appropriate appendix, or link to a paper containing the proof.

3.1 Black-Box Separation of Adaptive and Non-Adaptive PWPP – Pavel Hubáček

The complexity class PWPP (Polynomial Weak Pigeonhole Principle) [Jeř16] captures collision finding within TFNP. Its canonical complete problem COLLISION asks: given a shrinking circuit

$C : \{0, 1\}^n \rightarrow \{0, 1\}^{n-1}$, find distinct $x_1, x_2 \in \{0, 1\}^n$ such that $C(x_1) = C(x_2)$. PWPP consists of all total search problems many-one reducible to COLLISION.

Jeřábek [Jeř16] showed that PWPP is closed under *non-adaptive* Turing reductions, i.e. $P^{\parallel \text{PWPP}} = \text{PWPP}$: solving k independently prepared PWPP instances reduces to a single collision query. A basic structural question for TFNP subclasses is whether they remain closed under *adaptive* Turing reductions, where later oracle queries may depend on earlier answers [BJ12]. For several classes (e.g. PLS, PPA, PPAD) adaptive and non-adaptive oracle access coincide [BJ12], while for the related class PPP (Polynomial *strong* Pigeonhole Principle), Fleming et al. [FGPR24] proved a black-box separation showing it is not Turing-closed.

Together with Bolzano, in [Hub26] we resolve the analogous question for PWPP in the black-box setting by introducing a natural adaptive task NESTEDCOLLISION, suggested by Bolzano, which requires two dependent collision-finding steps.

After being given the problem, Bolzano produced the core construction and proof in four rounds of interaction. While the initial proof had flaws, after being instructed to adopt a different high-level strategy,¹ Bolzano delivered a mostly complete formal proof in four additional rounds, with only minor typographical errors and no significant logical gaps.

Formal results. The theorem proven in [Hub26] is the following.

Theorem 1 (Black-box PWPP is not Turing-closed [Hub26]). *In the decision-tree model, the search problem NESTEDCOLLISION admits no shallow COLLISION-formulation. Therefore, black-box PWPP is not closed under adaptive Turing reductions.*

The Bolzano research transcript is available at [Bol26d]. The complete proof appears in [Hub26].

3.2 Structural Results on Multi-Slope Tilings – Jan Grebík

Motivation and result The problem originated in the study of *translational monotilings* of $\mathbb{R}^d$, or less generally $\mathbb{Z}^d$, where the setup is the following. We are given a measurable set $\Omega \subseteq \mathbb{R}^d$ of finite positive Lebesgue measure and want to understand if there is a (necessarily uniformly discrete) set of translates $T \subseteq \mathbb{R}^d$ such that

$$\Omega \oplus T = \mathbb{R}^d,$$

where $\Omega \oplus T$ means that the collection of translates $\{\Omega + t\}_{t \in T}$ are disjoint up to null sets and cover all of $\mathbb{R}^d$. In this case, T is called a *tiling of $\mathbb{R}^d$ by Ω* , and Ω is called a *tile*. The definitions for translational monotilings of $\mathbb{Z}^d$ are analogous. Much of the recent development in the area [Bha20, GT21, GT24, GT25] have been driven by the so-called *periodic tiling conjecture*.

Conjecture 2 (Periodic tiling conjecture (PTC), [LW96]). *Let $\Omega \subseteq \mathbb{R}^d$ be a tile. Then there is a tiling T of $\mathbb{R}^d$ by Ω that is periodic, that is, $\{\gamma \in \mathbb{R}^d : \gamma + T = T\}$ contains a lattice.*

Unlike in the case of general tiling problems, where more tiles are allowed, PTC holds in $\mathbb{Z}^2$ which was proven by Bhattacharya [Bha20]. On the other hand Greenfeld and Tao [GT24] showed that PTC fails in $\mathbb{Z}^d$ (and thus $\mathbb{R}^d$) for large enough d . In $\mathbb{R}^2$ Kenyon [Ken92] showed that PTC holds for tiles that are topological disks, but the general case is widely open. Together with de Dios, Greenfeld and Madrid we investigated the case of tiles that are polygonal sets (possibly disconnected with holes) with edges being axes parallel that may have irrational lengths. We obtain the following general statement, providing a weak form of PTC.

¹Concretely: “Discard the case analysis strategy. Instead, prove that $\Pi \notin PWPP^O$ by establishing that, with overwhelming probability, the reduction circuit contains “useless” collisions that do not reveal a solution to Π .”

Theorem 3 ([dDGGM26]). *Let $\Omega \subseteq \mathbb{R}^2$ be an axes parallel polygonal tile. Then there is $k \in \mathbb{N}$ and a tiling $T = T_1 \sqcup \dots \sqcup T_k$ of $\mathbb{R}^2$ by Ω such that after possibly swapping the vertical and horizontal axes the following holds:*

1. T_i is periodic for every $1 \leq i \leq k$,
2. $\Omega \oplus T_i$ is a union of cosets of $\mathbb{R}(0, 1)$,
3. there is $\gamma \in \mathbb{R}^2 \setminus \{0\}$ such that $T + \gamma = T$, more specifically, if $k > 1$, then γ is of the form $\gamma = (0, \alpha)$ for some $\alpha > 0$.

Note that if $k = 1$ (or in the discrete case $\mathbb{Z}^2$), then (1) above would already imply that T is periodic, thus establishing the PTC for Ω . Our current techniques do not seem to give any information about the possible relation between T_i 's or the sets of the form $\Omega \oplus T_i$. This leads to the question of whether there exist interesting tilings as in Theorem 3. Without any restriction, the answer to this question is trivial, as one might consider the lattice tiling of $\mathbb{R}^2$ by a unit square $[0, 1] \times [0, 1]$ and construct for any $k \in \mathbb{N}$ a tiling T as above by shifting different columns. In order to avoid this trivial case we need a definition.

Definition 4. We say that $\Omega \subseteq \mathbb{R}^2$ *tils a column* (or is a *column tile*), if there is $S \subseteq \mathbb{R}(0, 1)$ such that $\Omega \oplus S$ is a union of cosets of $\mathbb{R}(0, 1)$.

Bolzano produced non-column examples for Theorem 3, where $T = T_1 \sqcup T_2$ and each part of the decomposition is periodic with different lattices. This is a first step towards understanding additional flexibility that tilings of $\mathbb{R}^2$ enjoy compared to tilings of $\mathbb{Z}^2$.

Theorem 5. *For every irrational $\alpha \in (2/3, 1)$ there is an axis parallel polygonal tile $\Omega_\alpha \subseteq \mathbb{R}^2$ that is not a column tile together with a tiling $T_\alpha = T_{1,\alpha} \sqcup T_{2,\alpha}$ of $\mathbb{R}^2$ by Ω_α such that the following holds:*

1. $T_{1,\alpha}$ is $(2\mathbb{Z}) \times \mathbb{Z}$ periodic,
2. $T_{2,\alpha}$ is periodic with lattice $\{(2k, k\alpha + \ell) : k, \ell \in \mathbb{Z}\}$,
3. $\Omega_\alpha \oplus T_{i,\alpha}$ is a union of cosets of $\mathbb{R}(0, 1)$ for $1 \leq i \leq 2$,
4. T_α is $(0, 1)$ -periodic.

The proofs appear in Section A. The Bolzano research transcript is available at [Bol26e].

Remark 6. Let us mention that each of the tiles Ω_α tile a horizontal column (that is, when we swap the horizontal and vertical axes, then the modified tile Ω'_α tiles a column), and the projection of $\Omega_\alpha \oplus T_i$ to the first coordinate is equal to $(2\mathbb{Z} + (i - 1)) + [0, 1]$. It seems that finding examples that would not have such a structure, or that would admit tilings that split as $T = T_1 \sqcup T_2 \sqcup T_3$ with the period of T_3 parametrized by $\beta > 0$ such that both $\beta, \beta - \alpha$ are irrational, is significantly more complicated, and Bolzano was not able to find an example nor to prove a general obstruction result for any of these conditions.

3.3 Special Soundness for Univariate KZG Batching – Pavel Hubáček

Polynomial Commitment Schemes (PCS) are a core building block of modern zero-knowledge proofs (zk-SNARKs). To minimize proof size and verification costs, practical systems often rely on batching techniques that allow a prover to aggregate evaluations of multiple polynomials at multiple points into a single proof. For the popular univariate KZG commitment scheme [KZG10],

existing multi-polynomial, multi-point batching protocols (e.g., [BDFG20]) have predominantly been analyzed only in idealized settings, limiting the assurance for their practical deployments. Proving their knowledge soundness in the standard model under falsifiable assumptions is a notoriously difficult task; the first prior standard-model analysis was strictly limited to the simpler case of batching evaluations of many polynomials at a *single* evaluation point [LPS25].

In the development of CHOPIN [BHKM26], an optimal pairing-based multilinear PCS, we required a fully rigorous standard-model security proof for multi-polynomial, multi-point batching that was not known. The human researchers proposed a variant of the KZG batch proof from [BDFG20] to simplify the task of proving its *special soundness in the standard model*, the core task towards a complete proof of knowledge soundness of the scheme. However, the human researchers did not have any rigorous proof.

Together with Bolzano, we resolved this gap. To assist the system, the input also contained a proof of the more basic theorem establishing the special soundness for batching KZG evaluation proofs for many polynomials at a single evaluation point from [LPS25], streamlined by the human researchers. After being given the problem, Bolzano found the proof of special soundness for the proposed variant of univariate KZG batching in six rounds of interaction. The proof was lengthy and technical, but, besides minor edits in notation and presentation, it was complete and formal as verified by the authors.

Formal results. The lemma proven in [BHKM26] is the following.

Lemma 7 (Special soundness of multi-polynomial, multi-point KZG batching [BHKM26, Lemma 3]). *Let $m, M, M' \in \text{poly}(\lambda)$. Let $T = \bigcup_{t=1}^m S_t$ and let $L = |T| + M$. Assume that KZG for degree less than M has M -special soundness in the following sense. From any set of $M' \geq M$ accepting KZG opening transcripts for the same commitment C at M' distinct points $a_1, \dots, a_{M'}$ with $a_j \neq \tau$, one can extract a polynomial $p \in \mathbb{F}[X]^{<M}$ such that $C = [p(\tau)]_1$ and $p(a_j) = y_j$ for all $j \in [M']$. Then there exists an extractor that, given as input any product-structured accepting (m, L) -tree $\mathcal{T} = (\text{tr}_{ij})_{i \in [m], j \in [L]}$ for the batch evaluation protocol of Figure 7 of [BHKM26], outputs polynomials $p_1, \dots, p_m \in \mathbb{F}[X]^{<M}$ such that $C_t = [p_t(\tau)]_1$ for all $t \in [m]$ and $p_t(z) = \eta_z$ for all $t \in [m]$ and all $z \in S_t$. Equivalently, the batch evaluation protocol is (m, L) -special sound with respect to product-structured accepting transcript trees.*

The Bolzano research transcript is available at [Bol26b]. The complete proof appears in [BHKM26].

3.4 Equivalence of Weak and Strong Working Set Properties for Heaps – Václav Rozhoň

The setting for this subsection is the beyond worst-case theory of data structures, in particular heaps. Two plausible definitions of a beyond worst-case heap occur in the literature – the strong working set property from [Iac00] and the weak working set property from [Elm06]. The strong working set property in particular has been crucial in a recent line of research [HHR⁺23, Iac00].

It was long assumed that the strong working set property was strictly stronger than the weak one. Surprisingly, they turn out to be equivalent. This was first proven by a human expert, but when given this task, Bolzano independently came up with a different proof and suggested a quantitative strengthening Equation (1) that was not apparent from the original proof. The strengthening gives an even tighter picture of how close the two definitions are.

Formal results Consider heaps that support the operations INSERT and EXTRACTMIN. We study two notions of locality-sensitive cost for such data structures.

Definition 8 (Weak working set property). A heap satisfies the *weak working set property* if, for any sequence of m operations, the total cost of serving them is $O(m + \sum_x \log(t'_x - t_x + 1))$, where the sum is over all extracted elements x , t_x is the insertion time of x , and t'_x is the extraction time of x .

Definition 9 (Strong working set property). A heap satisfies the *strong working set property* if, for any sequence of m operations, the total cost of serving them is $O(m + \sum_x \max_{t_x \leq t < t'_x} \log(|W_{t,x}| + 1))$, where $W_{t,x}$ is the set of all elements that have been inserted after time t_x and are still present at time t .

Theorem 10. *A heap has the weak working set property if and only if it has the strong one. In fact, the following holds for any $\varepsilon > 0$.*

$$\sum_x \log L_x \leq (1 + \varepsilon) \sum_x \log K_x + O(m/\varepsilon) \quad (1)$$

where $L_x = t'_x - t_x + 1$ is the lifetime of element x and $K_x = \max_{t_x \leq t < t'_x} (|W_{t,x}| + 1)$ is its strong working set cost.

The theorem is proven in Section B. The Bolzano research transcript is available at [Bol25a].

3.5 Partitioning under Function Preimage Constraints – Robert Šámal

KAMAK is a Czech problem-solving workshop where participants propose and collect open problems in combinatorics and discrete mathematics. The present problem appears as Problem 1 in the collection from the 2020 edition [Krá20]. The problems are suggested by the participants and are of various difficulty, but always meant as potentially interesting research problems.

Consider sets E and F , together with functions $f_1, \dots, f_k : E \rightarrow F$ satisfying *pointwise distinctness*, meaning that

$$f_i(x) \neq f_j(x) \quad \text{for every } x \in E \text{ and every } i \neq j.$$

Conjecture suggested by C. Feghali asked whether a condition on fibre sizes forces a certain bounded partition. (Pointwise distinctness is obviously necessary.) The case $k = 2$ was known and had applications in digraph coloring [BHB06].

Conjecture 11 (Original conjecture; the case $k = 2$ is known). *If for every $z \in F$ there exists $t \in \{1, \dots, k\}$ with $|f_t^{-1}(z)| \leq n$, then E can be partitioned into $2n + 1$ parts $E_1, \dots, E_{2n+1}$ such that*

$$f_p(E_i) \cap f_q(E_i) = \emptyset \quad \text{for every } i \text{ and every } p < q.$$

A convenient way to view the problem is through the *conflict graph* on vertex set E , in which distinct $x, y \in E$ are adjacent whenever $f_p(x) = f_q(y)$ for some $p \neq q$. Then a partition with the required disjointness property is exactly a proper coloring of this graph. Bolzano's first contribution was to observe that for $k \geq 3$ the original hypothesis can be satisfied vacuously: a dummy function may have empty fibers and thus the remaining functions are not controlled.

This leads to the following negative result.

Theorem 12 (Counterexample to the original conjecture). *For every $n \geq 1$ and every $M \geq 1$, there exist sets E, F and functions $f_1, f_2, f_3 : E \rightarrow F$ such that:*

1. $f_i(x) \neq f_j(x)$ for all $x \in E$ and all $i \neq j$;
2. for every $z \in F$ there exists $t \in \{1, 2, 3\}$ with $|f_t^{-1}(z)| \leq n$;
3. every partition of E satisfying

$$f_p(E_i) \cap f_q(E_i) = \emptyset \quad \text{for all } i \text{ and all } p < q$$

requires more than M parts.

The construction realizes the conflict graph as a shift graph, whose chromatic number is unbounded. The full proof appears in Appendix C.

The counterexample suggests that the right assumption is not to control single fibers, but to control them *pairwise*. This leads to the following notion, *also suggested by Bolzano*.

Definition 13 (Pairwise n -boundedness). The functions $f_1, \dots, f_k : E \rightarrow F$ are *pairwise n -bounded* if for every $z \in F$ and every pair $p \neq q$,

$$\min(|f_p^{-1}(z)|, |f_q^{-1}(z)|) \leq n.$$

Under this stronger hypothesis one gets a positive result.

Theorem 14 (Pairwise n -boundedness implies bounded partition). *Assume pointwise distinctness and pairwise n -boundedness. Then E can be partitioned into $2nk(k-1)+1$ parts $E_1, \dots, E_{2nk(k-1)+1}$ such that*

$$f_p(E_i) \cap f_q(E_i) = \emptyset \quad \text{for every } i \text{ and every } p < q.$$

The proof orients each conflict toward the endpoint whose side of the witnessing equality comes from a small fiber, thereby obtaining a bounded-indegree orientation of the conflict graph. This implies bounded degeneracy and hence bounded chromatic number. Appendix C contains the full proof, together with a stronger bound under uniform n -boundedness and complementary lower-bound constructions.

The Bolzano research transcript is available at [Bol26c].

3.6 Complexity of Optimization in KKOS Cultural Dynamics – Martin Koutecký

Kempe, Kleinberg, Oren, and Slivkins [KKOS16] introduced a model of cultural dynamics in which agents on a social network update their opinions to reduce disagreement with neighbors. In their *local model*, the equilibrium condition requires that neighboring agents in the support of a distribution experience equal “mass” – that is, the total weight in their closed neighborhood is the same.

It is natural (especially motivated by bribery-type viewpoints, see [FGKT22]) to consider the problem of finding a closest equilibrium x to a given (arbitrary) distribution y . Formally, given an undirected graph $G = (V, E)$ with adjacency matrix A' , set $A = A' + I$ (adding self-loops), and given an initial distribution $y \in \mathbb{R}_{\geq 0}^V$ with $\|y\|_1 = 1$ and a cost vector $c \in \mathbb{R}_{\geq 0}^V$, the task is to find a distribution $x \in \mathbb{R}_{\geq 0}^V$ with $\sum_v x_v = 1$ minimizing $\sum_v c_v |x_v - y_v|$ subject to the constraint that for every edge $uv \in E$, if $x_u, x_v > 0$ then $(Ax)_u = (Ax)_v$.

Bolzano worked on this problem autonomously over three rounds. In Round 1, Bolzano found a clean NP-hardness proof via a reduction from CLIQUE: adding a universal vertex forces any feasible support to be a clique, and the ℓ_1 cost becomes monotone in the clique size. In Round 2, Bolzano established membership in NP (via an LP-based polynomial certificate), corrected a small error in

the original problem formulation,² and proved structural results for chordal graphs and forests. In Round 3, Bolzano designed a polynomial-time $O(n^2)$ algorithm for forests via dynamic programming on dissociation sets.

Formal results.

Theorem 15 (NP-completeness). *Under the standard binary encoding of rational inputs, the decision version of the KKOS optimization problem is NP-complete. NP-hardness holds even with unit costs, positive rational y , and a universal vertex. The optimization problem is NP-hard.*

Proposition 16 (Forest characterization). *On a forest, the feasible supports are exactly the dissociation sets, i.e., the vertex sets S such that $G[S]$ has maximum degree at most 1.*

Theorem 17 (Polynomial-time algorithm for forests). *If G is a forest, the optimization problem can be solved in $O(n^2)$ time. The forest-restricted decision problem is in P .*

Remark 18. On chordal graphs, connected feasible supports must be cliques (Theorem 34), so feasible supports are exactly disjoint unions of cliques.

The proofs appear in Section D. The Bolzano research transcript is available at [Bol26a].

Acknowledgements

We thank Tomáš Gavenčíak and Vojtěch Rozhoň for helpful discussions and for their contributions to the development of the Bolzano system. JG, VR, and AZ were supported by the Czech Science Foundation (GA ČR), project No. 26-23599M. RŠ was supported by grant no.25-16627S from the Czech Science Foundation (GAČR). MK was supported by grant no.25-17221S from the Czech Science Foundation (GAČR).

References

- [ABH⁺26] Mohammed Abouzaid, Andrew J. Blumberg, Martin Hairer, Joe Kileel, Tamara G. Kolda, Paul D. Nelson, Daniel Spielman, Nikhil Srivastava, Rachel Ward, Shmuel Weinberger, and Lauren Williams. First Proof, 2026.
- [B⁺25] Benjamin Breen et al. Ax-Prover: A deep reasoning agentic framework for theorem proving in mathematics and quantum physics, 2025.
- [BCE⁺25] Sébastien Bubeck, Christian Coester, Ronen Eldan, Timothy Gowers, Yin Tat Lee, Alexandru Lupsasca, Mehtaab Sawhney, Robert Scherrer, Mark Sellke, Brian K. Spears, Derya Unutmaz, Kevin Weil, Steven Yin, and Nikita Zhivotovskiy. Early science acceleration experiments with GPT-5, 2025.
- [BDFG20] Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Efficient polynomial commitment schemes for multiple points and polynomials. *IACR Cryptol. ePrint Arch.*, page 81, 2020.

²The task statement asserted that for a fixed support S , one can compute a cost-minimizing vector by linear programming. This is not true in the straightforward LP formulation because the exact-support constraint involves strict inequalities, so the feasible region can be open. Bolzano provided a counterexample and a correct reformulation using a max-margin LP.

- [Bha20] Siddhartha Bhattacharya. Periodicity and decidability of tilings of $\mathbb{Z}^2$. *American Journal of Mathematics*, 142(1):255–266, 2020.
- [BHB06] S. Bessy, F. Havet, and E. Birmelé. Arc-chromatic number of digraphs in which every vertex has bounded outdegree or bounded indegree. *J. Graph Theory*, 53(4):315–332, December 2006.
- [BHKM26] Juraĭ Belohorec, Pavel Hubáček, Aleksĭ Kalsta, and Kristýna Mašková. CHOPIN: Optimal pairing-based multilinear polynomial commitments from bivariate KZG. *IACR Cryptol. ePrint Arch.*, page 480, 2026. <https://eprint.iacr.org/archive/2026/480/20260308:102759>.
- [BJ12] Samuel R. Buss and Alan S. Johnson. Propositional proofs and reductions between NP search problems. *Annals of Pure and Applied Logic*, 163(9):1163–1182, 2012.
- [Bol25a] Bolzano transcript: Heap equivalence. <https://bolzano.app/heap-equivalence>, 2025.
- [Bol25b] Bolzano Team. Bolzano. <https://bolzano.app/>, 2025.
- [Bol26a] Bolzano transcript: KKOS optimization. <https://bolzano.app/kkos-optimization>, 2026.
- [Bol26b] Bolzano transcript: KZG batching. <https://bolzano.app/kzg-batching>, 2026.
- [Bol26c] Bolzano transcript: Function-preimage partitioning. <https://bolzano.app/function-preimage-partitioning>, 2026.
- [Bol26d] Bolzano transcript: PWPP separation. <https://bolzano.app/pwpp-separation>, 2026.
- [Bol26e] Bolzano transcript: Multi-slope tilings. <https://bolzano.app/multi-slope-tilings>, 2026.
- [D⁺25] Jasper Dekoninck et al. The Open Proof Corpus: A large-scale study of LLM-generated mathematical proofs, 2025.
- [dDGGM26] Jaume de Dios Pont, Jan Grebík, Rachel Greenfeld, and José Madrid. Translational tilings by axes-parallel polygonal sets. Work in progress, 2026.
- [DLT⁺23] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023.
- [Elm06] Amr Elmasry. A priority queue with the working-set property. *Int. J. Found. Comput. Sci.*, 17(6):1455–1466, 2006.
- [F⁺26a] Tony Feng et al. Semi-autonomous mathematics discovery with Gemini: A case study on the Erdős problems, 2026.
- [F⁺26b] Tony Feng et al. Towards autonomous mathematics research, 2026.
- [FGKT22] Piotr Faliszewski, Rica Gonen, Martin Koutecký, and Nimrod Talmon. Opinion diffusion and campaigning on society graphs. *J. Log. Comput.*, 32(6):1162–1194, 2022.

- [FGPR24] Noah Fleming, Stefan Grosser, Toniann Pitassi, and Robert Robere. Black-box PPP is not Turing-closed. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1405–1414, 2024.
- [FJhK⁺26] Tony Feng, Junehyuk Jung, Sang hyun Kim, Carlo Pagano, et al. Aletheia tackles FirstProof autonomously, 2026.
- [G⁺25] Jiaxing Guo et al. Right is not enough: The pitfalls of outcome supervision in training LLMs for math reasoning, 2025.
- [GGSTW25] Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical exploration and discovery at scale, 2025.
- [GT21] Rachel Greenfeld and Terence Tao. The Structure of Translational Tilings in $\mathbb{Z}^d$. *Discrete Analysis*, 2021.
- [GT24] Rachel Greenfeld and Terence Tao. A counterexample to the periodic tiling conjecture. *Annals of Mathematics*, 200(1):301–363, 2024.
- [GT25] Rachel Greenfeld and Terence Tao. Undecidability of translational monotilings. *J. Eur. Math. Soc.*, 2025.
- [HHR⁺23] Bernhard Haeupler, Richard Hladík, Václav Rozhoň, Robert Tarjan, and Jakub Tětek. Universal optimality of Dijkstra via beyond-worst-case heaps, 2023.
- [HMS⁺25] Thomas Hubert, Rishi Mehta, Laurent Sartran, Miklós Z. Horváth, et al. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 2025.
- [Hub26] Pavel Hubáček. Black-box PWPP is not Turing closed. *CoRR*, abs/2602.23809, 2026. <https://doi.org/10.48550/arXiv.2602.23809>.
- [HY25] Hao Huang and Lin F. Yang. Winning gold at IMO 2025 with a model-agnostic verification-and-refinement pipeline, 2025.
- [Iac00] John Iacono. Improved upper bounds for pairing heaps. In *Scandinavian Workshop on Algorithm Theory*, pages 32–45. Springer, 2000.
- [Jeř16] Emil Jeřábek. Integer factoring and modular square roots. *Journal of Computer and System Sciences*, 82(2):380–394, 2016.
- [Ken92] Richard Kenyon. Rigidity of planar tilings. *Inventiones Mathematicae*, 107(3):637–651, 1992.
- [KKOS16] David Kempe, Jon M. Kleinberg, Sigal Oren, and Aleksandrs Slivkins. Selection and influence in cultural dynamics. *Netw. Sci.*, 4(1):1–27, 2016.
- [Krá20] Karel Král. Open problems. <https://kam.mff.cuni.cz/~kamak/static/problems/2020.pdf>, 2020. KAMAK Problem Solving Workshop.
- [KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 2010.

- [Log25] Logical Intelligence. Aleph Prover, 2025. <https://logicalintelligence.com/aleph-prover.html>.
- [LPS25] Helger Lipmaa, Roberto Parisella, and Janno Siim. On knowledge-soundness of Plonk in ROM from falsifiable assumptions. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part VII*, volume 16006 of *Lecture Notes in Computer Science*, pages 362–395. Springer, 2025.
- [LW96] Jeffrey C. Lagarias and Yang Wang. Tiling the line with translates of one tile. *Inventiones Mathematicae*, 124(1–3):341–365, 1996.
- [MTG⁺23] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.
- [N⁺25] Alexander Novikov et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery, 2025.
- [Ope26] OpenAI. Our First Proof submissions. <https://openai.com/index/first-proof-submissions/>, 2026.
- [PBF24] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations, 2024.
- [PSV26] Moe Putterman, Mehtaab Sawhney, and Gregory Valiant. On infinite sets with no 3 on a line, 2026.
- [RPBN⁺24] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.
- [TWL⁺24] Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.
- [Urb26] Josef Urban. 130k lines of formal topology in two weeks: Simple and cheap autoformalization for everyone?, 2026.
- [V⁺25] Sumanth Varambally et al. Hilbert: Recursively building formal proofs with informal reasoning, 2025.
- [WCAJ⁺26] David P. Woodruff, Vincent Cohen-Addad, Lalit Jain, Jieming Mao, Song Zuo, et al. Accelerating scientific research with Gemini: Case studies and common techniques, 2026.

A Structural Results on Multi-Slope Tilings

This appendix contains the proofs of Theorem 5 from Section 3.2.

The periodicity in Theorem 3 allows to simplify the problem by working in the group $G = \mathbb{Z} \times \mathbb{T}$, where $\mathbb{T} := \mathbb{R}/\mathbb{Z}$, with counting measure on $\mathbb{Z}$ and normalized Lebesgue measure μ on $\mathbb{R}/\mathbb{Z}$. Addition is componentwise, that is, $(m, \theta) + (n, \varphi) = (m+n, \theta + \varphi)$. All equalities, coverings, and disjointness statements are understood a.e. with respect to the product measure.

In this setup, we work with measurable sets $A \subset G$ of the form $A = \bigcup_{i=1}^{\ell} \{n_i\} \times I_i$ with $n_i \in \mathbb{Z}$ distinct and each $I_i \subset \mathbb{T}$ a half-open interval. The *integer support* of A is $\text{supp}(A) := \{n_i : 1 \leq i \leq \ell\}$, and the *fiber* at n is $A_n := \{t \in \mathbb{T} : (n, t) \in A\}$. A set $T \subset G$ is a *tiling of G by A* if $A + T = G$ and the translates $\{A + t\}_{t \in T}$ are pairwise disjoint up to null sets. This is denoted as $A \oplus T = G$. If $A \oplus T = G$, then A is called a *tile*.

Analogously to Definition 4, a set $A \subseteq G$ is a *column tile* if there exist finite sets $C \subset \mathbb{Z}$ and $\Lambda \subset \mathbb{T}$ such that $A \subset C \times \mathbb{T}$ and the vertical translates $\{A + (0, \lambda) : \lambda \in \Lambda\}$ tile $C \times \mathbb{T}$.

A.1 Auxiliary claims

Lemma 19 (Multi-coset affine arithmetic-splitting criterion). *Let $q \geq 1$, and $H \subset q\mathbb{Z}$ be a finite set of distinct integers. Let $J_h \subset \mathbb{T}$ be measurable for each $h \in H$, and $\beta_r, \alpha_r \in \mathbb{T}$ for each $0 \leq r \leq q-1$. Define*

$$A := \bigcup_{h \in H} (\{h\} \times J_h) \subset G \text{ and } T_r := \{(qk + r, \beta_r + k\alpha_r) : k \in \mathbb{Z}\} \subset G \text{ for each } 0 \leq r \leq q-1.$$

Then the following are equivalent:

1. $A \oplus T = G$ where $T = \bigcup_{r=0}^{q-1} T_r$,
2. for every $r \in R$, one has the circle partition identity

$$\bigsqcup_{h \in H} \left(J_h - \frac{h}{q} \alpha_r \right) = \mathbb{T} \quad \text{a.e.}$$

Proof. First observe that if $1 \leq r \leq q-1$ and $t = (qk + r, \beta_r + k\alpha_r) \in T_r$, then $A + t \subseteq (q\mathbb{Z} + r) \times \mathbb{T}$. Hence, it is enough to understand each fixed $1 \leq r \leq q-1$ separately.

For $m \in \mathbb{Z}$, define the set

$$F_{r,m} = \{\theta \in \mathbb{T} : (r + qm, \theta) \in A + T_r\}.$$

A point $(r + qm, \theta)$ lies in $A + (qk + r, \beta_r + k\alpha_r)$ for some $k \in \mathbb{Z}$ if and only if there exists $h \in H$ such that

- $q(m - k) = h$, and
- $\theta \in J_h + \beta_r + k\alpha_r$.

Because $h \in q\mathbb{Z}$, the above equation has the unique solution $k = m - \frac{h}{q}$. Substituting this into the vertical condition gives $\theta \in \left(J_h - \frac{h}{q} \alpha_r \right) + \beta_r + m\alpha_r$. This gives

$$F_{r,m} = \left(\bigcup_{h \in H} \left(J_h - \frac{h}{q} \alpha_r \right) \right) + \beta_r + m\alpha_r. \quad (2)$$

Now we are ready to show the equivalence.

(2) $\Rightarrow$ (1). Let $1 \leq r \leq q-1$ and $m \in \mathbb{Z}$. Since translation on $\mathbb{T}$ preserves Lebesgue measure and preserves pairwise disjointness a.e., the assumed partition identity implies $F_{r,m} = \mathbb{T}$ a.e., and the pieces contributing to (2) are pairwise disjoint a.e. As $m \in \mathbb{Z}$ was arbitrary, we get that $A \oplus T_r = (q\mathbb{Z} + r) \times \mathbb{T}$. This proves (1) by the remark in the beginning of this proof.

(1) $\Rightarrow$ (2). By the remark in the beginning of the proof and the assumption, we have that (2) reads as

$$\mathbb{T} = F_{r,0} = \left(\bigsqcup_{h \in H} \left(J_h - \frac{h}{q} \alpha_r \right) \right) + \beta_r.$$

for every $1 \leq r \leq q-1$ and $m = 0$. Shifting by $-\beta_r$ gives (2). $\square$

Proposition 20 (Fiberwise criterion for column tiles). *Let $A = \bigcup_{i=1}^{\ell} \{n_i\} \times I_i \subseteq \mathbb{Z} \times \mathbb{T}$ be a column tile. Then there is $k \in \mathbb{N}$ such that*

$$\mu(A_{n_i}) = 1/k$$

for every $1 \leq i \leq \ell$.

Proof. By the definition, there are $C \subset \mathbb{Z}$ and a finite nonempty set $\Lambda \subset \mathbb{T}$ such that the translates $\{A + (0, \lambda) : \lambda \in \Lambda\}$ are pairwise disjoint a.e., and $\bigsqcup_{\lambda \in \Lambda} (A + (0, \lambda)) = C \times \mathbb{T}$.

For each $1 \leq i \leq \ell$, observe that $\mathbb{T} = \bigsqcup_{\lambda \in \Lambda} A_{n_i} + \lambda$. Therefore

$$1 = \mu(\mathbb{T}) = \sum_{\lambda \in \Lambda} \mu(A_{n_i} + \lambda) = |\Lambda| \mu(A_{n_i})$$

as Lebesgue measure is translation-invariant. $\square$

A.2 Explicit construction

Choose an irrational $\varepsilon \in (0, 1/3)$ and define:

- $\alpha = 1 - \varepsilon$,
- $I_0 := [\varepsilon, 2\varepsilon)$, $I_2 := [2\varepsilon, 1)$, $I_4 := [0, \varepsilon)$,
- $A_\alpha = (\{0\} \times I_0) \cup (\{2\} \times I_2) \cup (\{4\} \times I_4) \subset \mathbb{Z} \times \mathbb{T}$,
- $S_{1,\alpha} := \{(2k, 0) : k \in \mathbb{Z}\}$ and $S_{2,\alpha} := \{(2k+1, k\alpha) : k \in \mathbb{Z}\}$.

We show that $A_\alpha \oplus S_\alpha = G$ where $S_\alpha = S_{1,\alpha} \sqcup S_{2,\alpha}$. We verify (2) in Lemma 19 with $q = 2$, $\beta_0 = \beta_1 = 0$, $\alpha_0 = 0$ and $\alpha_1 = \alpha$. For $r = 0$, we have

$$\bigsqcup_{2i \in \{0, 2, 4\}} I_{2i} = [\varepsilon, 2\varepsilon) \sqcup [2\varepsilon, 1) \sqcup [0, \varepsilon) = \mathbb{T},$$

and for $r = 1$, we have

$$\begin{aligned} \bigsqcup_{2i \in \{0, 2, 4\}} (I_{2i} - i\alpha) &= [\varepsilon, 2\varepsilon) \sqcup ([2\varepsilon, 1) - (1 - \varepsilon)) \sqcup ([0, \varepsilon) - 2(1 - \varepsilon)) \\ &= [\varepsilon, 2\varepsilon) \sqcup ([0, \varepsilon) \sqcup [3\varepsilon, 1)) \sqcup ([2\varepsilon, 3\varepsilon)) \\ &= \mathbb{T}, \end{aligned}$$

where $[2\varepsilon, 1) - (1 - \varepsilon) = [0, \varepsilon) \sqcup [3\varepsilon, 1)$ follows from the assumption that $\varepsilon \in (0, 1/3)$.

Observe that, by Proposition 20, A is not a column tile as $\mu(A \cap (\{0\} \times \mathbb{T}))$ is not rational.

A.3 Proof of Theorem 5

Given an irrational $\alpha \in (2/3, 1)$, consider the tile $A_\alpha \subseteq \mathbb{Z} \times \mathbb{T}$ described above, and define

$$\Omega_\alpha = \{(n, x) + (t, 0) : x, t \in [0, 1) \text{ and } (n, x) \in A_\alpha\} \subseteq \mathbb{R}^2.$$

It follows from the properties of A_α and S_α that Ω_α does not tile a column, and that $T_\alpha = T_{1,\alpha} \sqcup T_{2,\alpha}$, where $T_{1,\alpha} = (2\mathbb{Z}) \times \mathbb{Z}$ and $T_{2,\alpha} = \{(2k, k\alpha + \ell) : k, \ell \in \mathbb{Z}\}$, have all the desired properties.

B Equivalence of Weak and Strong Working Set Properties

In this appendix we prove the equivalence between the weak and strong working set properties stated in Section 3.4. In fact, we prove the stronger quantitative estimate

$$\sum_x \log L_x \leq (1 + \varepsilon) \sum_x \log K_x + O(m/\varepsilon), \quad (3)$$

valid for every $\varepsilon > 0$, where the sum ranges over all extracted elements.

The easy point is that $K_x \leq L_x$ for every element x , so the strong working set bound always implies the weak one. The substance is the converse direction: although an individual lifetime L_x can be much larger than the corresponding strong parameter K_x , this can only happen for relatively few elements at the same time. The proof makes this precise via a packing argument: at any time t , there cannot be many live elements with small strong parameter. Grouping elements according to the size of K_x , this packing bound yields an upper bound on the total lifetime within each group, and Jensen's inequality then controls the total contribution of $\log(L_x/K_x)$.

Setup. Consider any sequence of m heap operations, each either INSERT of a fresh element or EXTRACTMIN removing a present element. Elements are distinct even if keys coincide. For each element x that is eventually extracted, let t_x be its insertion time, t'_x its extraction time, and define its *lifetime*

$$L_x := t'_x - t_x + 1.$$

For each time t with $t_x \leq t < t'_x$, let $W_{t,x}$ be the set of all elements inserted after time t_x that are still present immediately after operation t . We also define

$$K_x := \max_{t_x \leq t < t'_x} (|W_{t,x}| + 1).$$

Thus L_x is the weak working set parameter and K_x is the strong one.

For a time t , let $A(t)$ denote the set of elements present immediately after operation t . All logarithms are base 2.

Lemma 21. *For every extracted element x , we have $K_x \leq L_x$.*

Proof. Fix t with $t_x \leq t < t'_x$. Every element of $W_{t,x}$ was inserted after time t_x and no later than time t , so $|W_{t,x}| \leq t - t_x$. Hence

$$|W_{t,x}| + 1 \leq (t - t_x) + 1 \leq (t'_x - t_x) + 1 = L_x.$$

Taking the maximum over all admissible t gives $K_x \leq L_x$. $\square$

The next lemma is the key combinatorial input. It says that elements with small strong parameter cannot overlap too much in time.

Lemma 22 (Packing lemma). *Fix a time t and an integer $k \geq 1$. Then at most k elements $x \in A(t)$ satisfy $K_x \leq k$.*

Proof. Suppose for contradiction that there are $k + 1$ such elements alive after operation t . Let x be the oldest among them, i.e. the one with minimum insertion time. Then the other k elements were all inserted after x and are all present at time t , hence they all belong to $W_{t,x}$. Therefore

$$|W_{t,x}| \geq k, \quad \text{so} \quad K_x \geq |W_{t,x}| + 1 \geq k + 1,$$

contradicting $K_x \leq k$. □

We now prove the quantitative comparison (3).

Theorem 23. *For every $\varepsilon > 0$ and every operation sequence of length m ,*

$$\sum_x \log L_x \leq (1 + \varepsilon) \sum_x \log K_x + O(m/\varepsilon),$$

where the sum ranges over all extracted elements x .

Proof. Fix $\varepsilon > 0$. We may assume $\varepsilon \leq 1$, since otherwise the claim is weaker than the case $\varepsilon = 1$. Let

$$b := \left\lceil \frac{1}{\varepsilon} \right\rceil,$$

so that $1/b \leq \varepsilon$ and $b = O(1/\varepsilon)$.

We partition the extracted elements into coarse levels according to the size of K_x : for each integer $j \geq 0$, let

$$X_j := \{x : 2^{bj} \leq K_x < 2^{b(j+1)}\}, \quad N_j := |X_j|.$$

We first bound the total lifetime inside one level. Since an element x is alive after exactly the times $t = t_x, t_x + 1, \dots, t'_x - 1$, it contributes $L_x - 1$ to $\sum_t \mathbf{1}_{x \in A(t)}$. Therefore

$$\sum_{x \in X_j} (L_x - 1) = \sum_{t=1}^m |X_j \cap A(t)|.$$

Now if $x \in X_j$, then $K_x < 2^{b(j+1)}$, hence $K_x \leq 2^{b(j+1)} - 1$. Applying Lemma 22 with $k = 2^{b(j+1)} - 1$, we get

$$|X_j \cap A(t)| \leq 2^{b(j+1)} - 1 \leq 2^{b(j+1)} \quad \text{for every } t.$$

Summing over t yields

$$\sum_{x \in X_j} (L_x - 1) \leq m \cdot 2^{b(j+1)},$$

and hence

$$\sum_{x \in X_j} L_x \leq m \cdot 2^{b(j+1)} + N_j \leq 2m \cdot 2^{b(j+1)}. \tag{4}$$

We now estimate the excess

$$G := \sum_x \log \left(\frac{L_x}{K_x} \right).$$

For $x \in X_j$ we have $K_x \geq 2^{bj}$, and therefore

$$\log \left(\frac{L_x}{K_x} \right) \leq \log \left(\frac{L_x}{2^{bj}} \right).$$

Set $y_x := L_x/2^{bj}$. By Lemma 21, we have $y_x \geq 1$. Also, by (4),

$$\sum_{x \in X_j} y_x = \sum_{x \in X_j} \frac{L_x}{2^{bj}} \leq \frac{2m \cdot 2^{b(j+1)}}{2^{bj}} = 2m \cdot 2^b.$$

Since $\log$ is concave, Jensen's inequality gives

$$\sum_{x \in X_j} \log\left(\frac{L_x}{K_x}\right) \leq \sum_{x \in X_j} \log y_x \leq N_j \log\left(\frac{2m \cdot 2^b}{N_j}\right).$$

Summing over all levels,

$$G \leq \sum_j N_j \log\left(\frac{2m \cdot 2^b}{N_j}\right).$$

Let $n := \sum_j N_j$ be the number of extracted elements, and let $p_j := N_j/n$. Then

$$G \leq n \log\left(\frac{2m \cdot 2^b}{n}\right) + nH(p),$$

where $H(p) := -\sum_j p_j \log p_j$ is the entropy of the distribution $(p_j)_j$.

We bound the two terms separately.

For the first term,

$$n \log\left(\frac{2m \cdot 2^b}{n}\right) = n(b+1) + n \log(m/n).$$

Since $n \leq m$, the first summand is at most $m(b+1)$. For the second, using the elementary inequality $\log z \leq (z-1)/\ln 2$ for $z \geq 1$, we get

$$n \log(m/n) \leq \frac{m}{\ln 2}.$$

Hence

$$n \log\left(\frac{2m \cdot 2^b}{n}\right) = O(mb). \tag{5}$$

For the entropy term, compare p with the geometric distribution $q_j := 2^{-(j+1)}$ on $\{0, 1, 2, \dots\}$. Nonnegativity of relative entropy gives

$$H(p) \leq -\sum_j p_j \log q_j = \sum_j p_j(j+1) = 1 + \sum_j p_j j,$$

and thus

$$nH(p) \leq n + \sum_j N_j j.$$

Now if $x \in X_j$, then $K_x \geq 2^{bj}$, so $\log K_x \geq bj$, i.e.

$$j \leq \frac{\log K_x}{b}.$$

Therefore

$$\sum_j N_j j = \sum_x j(x) \leq \frac{1}{b} \sum_x \log K_x,$$

where $j(x)$ denotes the unique level index such that $x \in X_{j(x)}$. Since also $n \leq m$, we obtain

$$nH(p) \leq m + \frac{1}{b} \sum_x \log K_x. \quad (6)$$

Combining (5) and (6), we conclude that

$$G \leq O(mb) + \frac{1}{b} \sum_x \log K_x.$$

Using $1/b \leq \varepsilon$ and $b = O(1/\varepsilon)$, this becomes

$$G \leq \varepsilon \sum_x \log K_x + O(m/\varepsilon).$$

Finally,

$$\sum_x \log L_x = \sum_x \log K_x + G \leq (1 + \varepsilon) \sum_x \log K_x + O(m/\varepsilon),$$

as claimed. $\square$

As an immediate consequence, the weak and strong working set properties are equivalent.

Corollary 24. *A heap satisfies the weak working set property if and only if it satisfies the strong working set property.*

Proof. If a heap satisfies the weak working set property, then its total cost on every operation sequence is

$$O\left(m + \sum_x \log L_x\right).$$

Applying Theorem 23 with any fixed ε (say $\varepsilon = 1$), we get

$$\sum_x \log L_x = O\left(m + \sum_x \log K_x\right),$$

and therefore the total cost is

$$O\left(m + \sum_x \log K_x\right),$$

which is exactly the strong working set property.

Conversely, if a heap satisfies the strong working set property, then by Lemma 21 we have $K_x \leq L_x$ for every extracted element x , hence

$$\sum_x \log K_x \leq \sum_x \log L_x.$$

So a bound of the form

$$O\left(m + \sum_x \log K_x\right)$$

immediately implies

$$O\left(m + \sum_x \log L_x\right),$$

which is the weak working set property. $\square$

C Partitioning under Function Preimage Constraints

In this appendix we prove the two main results stated in Section 3.5: the counterexample to the original conjecture (Theorem 12) and the positive partition theorem under the stronger pairwise boundedness hypothesis (Theorem 14). We also record two complementary observations: under the stronger assumption of uniform n -boundedness the constant improves from $2nk(k-1)+1$ to $nk(k-1)+1$, and even under pairwise n -boundedness one cannot hope for a bound smaller than $2k-1$ in general.

The natural language for the problem is graph coloring. The following reformulation will be used throughout.

Lemma 25 (Conflict graph reformulation). *Let E and F be sets, and let $f_1, \dots, f_k : E \rightarrow F$ satisfy $f_i(x) \neq f_j(x)$ for every $x \in E$ and every $i \neq j$. Define the conflict graph G on vertex set E by declaring distinct $x, y \in E$ adjacent whenever*

$$f_p(x) = f_q(y) \quad \text{for some } p \neq q.$$

Then for every integer $m \geq 1$, the following are equivalent:

1. *E admits a partition $E = E_1 \sqcup \dots \sqcup E_m$ such that*

$$f_p(E_i) \cap f_q(E_i) = \emptyset \quad \text{for every } i \text{ and every } p < q;$$

2. *the graph G is properly m -colorable.*

Proof. Suppose first that $E = E_1 \sqcup \dots \sqcup E_m$ is such a partition. If two distinct vertices $x, y \in E_i$ were adjacent in G , then for some $p \neq q$ we would have $f_p(x) = f_q(y)$, and hence $f_p(E_i) \cap f_q(E_i) \neq \emptyset$, a contradiction. Thus every E_i is an independent set, so the partition defines a proper m -coloring of G .

Conversely, suppose that $c : E \rightarrow \{1, \dots, m\}$ is a proper m -coloring of G , and let $E_i := c^{-1}(i)$. If for some i and some $p \neq q$ the sets $f_p(E_i)$ and $f_q(E_i)$ were not disjoint, we could choose $x, y \in E_i$ with $f_p(x) = f_q(y)$. Then $x \neq y$ by assumptions, so x and y would be adjacent in G , contradicting the fact that E_i is a color class. $\square$

C.1 A counterexample to the original conjecture

We now prove Theorem 12. The mechanism behind the counterexample is simple: while for $k = 2$ the assumption implies that the maximum degree in the conflict graph is at most $2n$, for $k \geq 3$, the original assumption can be satisfied vacuously by means of a “dummy” function with many empty fibers, while the remaining two functions create a highly chromatic conflict graph.

Proof of Theorem 12. Fix $n \geq 1$ and $M \geq 1$. Let $m > 2^M$, and define

$$E := \{(i, j) : 1 \leq i < j \leq m\}, \quad F := \{0, 1, \dots, m\}.$$

Define three functions on E by

$$f_1(i, j) := i, \quad f_2(i, j) := j, \quad f_3(i, j) := 0.$$

The pointwise distinctness condition holds: for every $(i, j) \in E$ we have $i < j$ and $0 \notin \{1, \dots, m\}$, so the three values $f_1(i, j), f_2(i, j), f_3(i, j)$ are pairwise distinct.

The fiber size condition is also satisfied. Indeed, if $z \in \{1, \dots, m\}$, then $f_3^{-1}(z) = \emptyset$, while for $z = 0$ we have $f_1^{-1}(0) = \emptyset$. Thus for every $z \in F$ there exists some $t \in \{1, 2, 3\}$ with $|f_t^{-1}(z)| = 0 \leq n$.

It remains to analyze the conflict graph G . Since f_3 takes only the value 0, while neither f_1 nor f_2 ever takes the value 0, no edge of G is witnessed by an equality involving f_3 . Thus two distinct vertices (i, j) and (i', j') are adjacent if and only if either

$$f_2(i, j) = f_1(i', j') \iff j = i',$$

or

$$f_1(i, j) = f_2(i', j') \iff i = j'.$$

This graph is known as the shift graph S_m . It is well-known, that $\chi(S_m) > M$; we include a simple proof to keep the treatment self-contained.

Let c be any proper coloring of S_m using r colors. For each $j \in \{1, \dots, m\}$ define

$$A_j := \{\ell : \text{there exists } i < j \text{ with } c(i, j) = \ell\}.$$

We claim that the sets $A_1, \dots, A_m$ are pairwise distinct subsets of $\{1, \dots, r\}$. Indeed, take $1 \leq j < k \leq m$, and let $\ell := c(j, k)$. Then $\ell \in A_k$ by definition. If also $\ell \in A_j$, then there exists $i < j$ with $c(i, j) = \ell$. But in the shift graph the vertices (i, j) and (j, k) are adjacent, since their second and first coordinates coincide, a contradiction, as c is a coloring. Hence $\ell \notin A_j$, so $A_j \neq A_k$.

Thus we have m distinct subsets of an r -element set, which implies $m \leq 2^r$. Since $m > 2^M$, we conclude that $r > M$. Therefore $\chi(S_m) > M$, and hence the conflict graph G is not M -colorable. By Lemma 25, any partition satisfying the required disjointness condition must use more than M parts. $\square$

The same construction immediately rules out the original conjecture for all larger values of k .

Corollary 26. *For every $k \geq 3$, the original conjecture fails.*

Proof. Starting from the construction above for f_1, f_2, f_3 , add functions $f_4, \dots, f_k$ whose ranges are disjoint from one another and from the range of f_1, f_2, f_3 . This preserves pointwise distinctness and does not remove any edges from the conflict graph, so the graph still contains the shift graph S_m and therefore has arbitrarily large chromatic number. $\square$

C.2 The strengthened hypothesis

The counterexample shows that the original assumption is too weak because it constrains only individual fibers. A natural fix is to require smallness *pairwise* across the functions involved in a potential conflict. We now prove Theorem 14.

Proof of Theorem 14. Let G be the conflict graph from Lemma 25; we orient its edges so that every vertex has bounded indegree. Consider an edge xy of G . By definition, there exist indices $p \neq q$ such that

$$f_p(x) = f_q(y) =: z.$$

By pairwise n -boundedness,

$$\min(|f_p^{-1}(z)|, |f_q^{-1}(z)|) \leq n.$$

If $|f_p^{-1}(z)| \leq n$, orient the edge from x to y ; otherwise orient it from y to x . (If both inequalities hold, choose either orientation.) We claim that every vertex has indegree at most $d := nk(k-1)$. Fix a vertex $y \in E$. For an incoming edge $x \rightarrow y$, there exist indices $p \neq q$ such that

$$f_p(x) = f_q(y) \quad \text{and} \quad |f_p^{-1}(f_q(y))| \leq n.$$

For any $y \in E$, the number of such triples (p, q, x) is at most $k(k-1)n$, thus $\text{indeg}(y) \leq k(k-1)n = d$.

Now any induced subgraph H of G has at most $d|V(H)|$ edges (counting arcs by their tails), thus average degree at most $2d$. Therefore, G is $2d$ -degenerate and therefore $(2d+1)$ -colorable, so

$$\chi(G) \leq 2nk(k-1) + 1.$$

The desired partition now follows from Lemma 25. $\square$

Under even stronger hypothesis that every fiber of every function has size at most n , one gets a better constant, because then the conflict graph has maximum degree (rather than merely maximum indegree) bounded by d .

Theorem 27 (Uniform n -boundedness). *Assume pointwise distinctness and suppose that*

$$|f_i^{-1}(z)| \leq n \quad \text{for every } i \in \{1, \dots, k\} \text{ and every } z \in F.$$

Then E can be partitioned into $nk(k-1) + 1$ parts with the required disjointness property.

Proof. Let G be the conflict graph. Fix a vertex $y \in E$. For each ordered pair (p, q) with $p \neq q$, define

$$N_{p,q}(y) := \{x \in E \setminus \{y\} : f_p(x) = f_q(y)\}.$$

Then $|N_{p,q}(y)| \leq |f_p^{-1}(f_q(y))| \leq n$. Since every neighbor of y belongs to at least one such set,

$$\deg_G(y) \leq \sum_{p \neq q} |N_{p,q}(y)| \leq nk(k-1).$$

Thus $\Delta(G) \leq nk(k-1)$, and therefore G is $(nk(k-1) + 1)$ -colorable. The claim follows from Lemma 25. $\square$

C.3 Lower bounds

The upper bounds above are not tight in general, but even under pairwise n -boundedness one cannot hope for a bound independent of k .

Proposition 28. *For every $k \geq 2$ and every $n \geq 1$, there exist examples satisfying pairwise n -boundedness for which at least $2k-1$ parts are required.*

Proof. Let $E = F = \mathbb{Z}_{2k-1}$ and define for $i = 1, \dots, k$

$$f_i(x) := x + i.$$

Each f_i is a bijection, so in particular all fibers have size $1 \leq n$. Hence the instance satisfies the stronger uniform n -boundedness condition.

We claim that the conflict graph is the complete graph K_{2k-1} . Indeed, let $x \neq y$ and write $\delta := y - x \in \mathbb{Z}_{2k-1} \setminus \{0\}$. The set of ordered differences $\{p - q : p \neq q\} = \{-(k-1), \dots, -1, 1, \dots, k-1\}$ coincides with all nonzero residues modulo $2k-1$. Hence there exist $p \neq q$ with $p - q = \delta$, which is equivalent to

$$f_p(x) = x + p = y + q = f_q(y).$$

Thus every two distinct vertices are adjacent. Therefore the conflict graph is K_{2k-1} and requires $2k-1$ colors. $\square$

D Complexity of Optimization in KKOS Cultural Dynamics

In this appendix we prove the results stated in Section 3.6. We first establish NP-hardness of the decision problem via a reduction from CLIQUE, then prove membership in NP under rational input encoding, yielding NP-completeness. Finally, we characterize feasible supports on forests and give a polynomial-time algorithm.

Throughout, $G = (V, E)$ is an undirected graph, A' is its adjacency matrix, $A = A' + I$ (self-loops added), $y \in \mathbb{R}_{\geq 0}^V$ is a distribution with $\|y\|_1 = 1$, and $c \in \mathbb{R}_{\geq 0}^V$ is a cost vector. A distribution $x \in \mathbb{R}_{\geq 0}^V$ with $\sum_v x_v = 1$ is *feasible* if for every edge $uv \in E$, whenever $x_u, x_v > 0$ one has $(Ax)_u = (Ax)_v$. We write $\text{supp}(x) = \{v : x_v > 0\}$ and call $(Ax)_v$ the *mass* at vertex v .

D.1 NP-hardness

Lemma 29 (Universal vertex forces clique support). *Let G be a graph with a universal vertex z . If x is a feasible distribution with $z \in \text{supp}(x)$, then $G[\text{supp}(x)]$ is a clique.*

Proof. Let $S = \text{supp}(x)$. Fix any $u \in S$ with $u \neq z$. Since z is universal, $zu \in E$. Because $x_z, x_u > 0$, feasibility gives $(Ax)_u = (Ax)_z$. Now z is adjacent to every vertex of S , and A includes self-loops, so $(Ax)_z = \sum_{v \in S} x_v = 1$. Hence $(Ax)_u = 1$. But $(Ax)_u = \sum_{v \in S \cap N_G[u]} x_v$, and every x_v with $v \in S$ is positive. This sum equals 1 = $\sum_{v \in S} x_v$ only if $S \subseteq N_G[u]$. Since this holds for every $u \in S$, the support induces a clique. $\square$

Lemma 30 (ℓ_1 cost lower bound). *Assume $c_i = 1$ for all i . Let $S \subseteq V$, and let x be any distribution with $\text{supp}(x) \subseteq S$. Then $\|x - y\|_1 \geq 2(1 - y(S))$, where $y(S) = \sum_{i \in S} y_i$.*

Proof. Because $x_i = 0$ for $i \notin S$, $\sum_{i \notin S} |x_i - y_i| = \sum_{i \notin S} y_i = 1 - y(S)$. Also, $\sum_{i \in S} x_i = 1$ and $\sum_{i \in S} y_i = y(S)$, so $\sum_{i \in S} (x_i - y_i) = 1 - y(S)$. By the triangle inequality, $\sum_{i \in S} |x_i - y_i| \geq 1 - y(S)$. Adding gives $\|x - y\|_1 \geq 2(1 - y(S))$. $\square$

Theorem 31 (NP-hardness). *The decision problem—given G , rational y, c , and rational threshold B , decide whether a feasible distribution x with $\sum_v c_v |x_v - y_v| \leq B$ exists—is NP-hard. Hardness holds even with $c_i = 1$ for all i , positive rational y , and G having a universal vertex.*

Proof. We reduce from CLIQUE. Let (H, k) be an instance with $H = (U, F)$ and $m = |U|$. Construct G by adding a universal vertex z adjacent to all of U . Set $c_i = 1$ for all i , $y_z = 2/3$, $y_u = 1/(3m)$ for each $u \in U$, and threshold $T_k = 2/3 - 2k/(3m)$.

Forward. If H has a clique C of size k , set $S = \{z\} \cup C$. Then $G[S]$ is a clique, so any distribution on S is feasible. Define $x_u = 1/(3m)$ for $u \in C$, $x_u = 0$ for $u \in U \setminus C$, $x_z = 1 - k/(3m)$. Its cost is $2(m - k)/(3m) = T_k$.

Backward. Suppose x is feasible with cost $\leq T_k$. If $x_z = 0$, the z -coordinate contributes $2/3$ and $\sum_{u \in U} |x_u - y_u| \geq 2/3$ (since $\sum_u x_u = 1$ while $\sum_u y_u = 1/3$), giving cost $\geq 4/3 > T_k$, a contradiction. So $z \in \text{supp}(x)$, and by Lemma 29, $\text{supp}(x) = \{z\} \cup C$ for a clique C in H of size t . By Lemma 30, the cost is $\geq 2/3 - 2t/(3m)$, so $t \geq k$. $\square$

D.2 Membership in NP

Theorem 32 (NP-membership). *Under the standard binary encoding of rational input, the decision problem belongs to NP.*

Proof. Let the instance be a yes-instance, with feasible distribution x and $S = \text{supp}(x)$. Consider the linear program $P(S)$ with variables $u \in \mathbb{R}^V$, $t \in \mathbb{R}^V$, $\delta \in \mathbb{R}$:

$$\begin{aligned}
u_i &= 0 && \text{for } i \notin S, \\
u_i &\geq \delta && \text{for } i \in S, \\
0 \leq u_i &\leq 1 && \text{for all } i, \quad \sum_i u_i = 1, \\
(Au)_a &= (Au)_b && \text{for every edge } ab \text{ with } a, b \in S, \\
t_i &\geq u_i - y_i, \quad t_i \geq y_i - u_i && \text{for all } i, \quad 0 \leq t_i \leq 1, \\
\sum_i c_i t_i &\leq B, && 0 \leq \delta \leq 1.
\end{aligned}$$

Maximize δ . The point $(u, t, \delta) = (x, |x - y|, \min_{i \in S} x_i)$ is feasible with $\delta > 0$, so the optimum is positive.

All coefficients are rational. An optimal basic feasible solution has coordinates whose length is polynomial in the input, so there exists an optimal solution (u^*, t^*, δ^*) with $\delta^* > 0$, $\text{supp}(u^*) = S$, and polynomial encoding length. Since u^* satisfies all constraints, it is a feasible distribution with cost $\leq B$. Thus every yes-instance has a polynomial-size rational certificate, verifiable in polynomial time. $\square$

D.3 Structural characterization

Lemma 33 (Dominated neighborhoods obstruct feasibility). *Let H be a graph. If there is an edge $uv \in E(H)$ with $N_H[u] \subsetneq N_H[v]$, then no strictly positive vector $x \in \mathbb{R}_{>0}^{V(H)}$ can satisfy $(Ax)_u = (Ax)_v$ for all edges.*

Proof. We have $(Ax)_v - (Ax)_u = \sum_{w \in N_H[v] \setminus N_H[u]} x_w > 0$, since $N_H[v] \setminus N_H[u] \neq \emptyset$ and $x_w > 0$ for all w . $\square$

Proposition 34 (Chordal feasible supports are disjoint unions of cliques). *Let G be a chordal graph. If $S \subseteq V$ is a feasible support, then every connected component of $G[S]$ is a clique.*

Proof. Let H be a connected component of $G[S]$. Since G is chordal, H is chordal. Restricting a feasible distribution to the vertices of H gives a strictly positive vector satisfying the mass-equality constraints on H .

If H is not a clique, then H has a simplicial vertex s that is not universal (every chordal graph has a simplicial vertex, and a non-clique has one that is not adjacent to all others). Pick any neighbor v of s in H . Since s is simplicial, $N_H(s)$ is a clique, so every neighbor of s is also a neighbor of v . Thus $N_H[s] \subseteq N_H[v]$, and since s is not universal, $N_H[s] \subsetneq N_H[v]$. By Lemma 33, no strictly positive feasible vector exists on H , a contradiction. $\square$

Proposition 35 (Feasible supports on forests are dissociation sets). *Let G be a forest and $S \subseteq V$ nonempty. Then there exists a feasible distribution with support exactly S if and only if $G[S]$ has maximum degree at most 1.*

Proof. ($\Rightarrow$) Suppose x is feasible with $\text{supp}(x) = S$. Let $H = G[S]$. If H has a component with ≥ 3 vertices, let u be a leaf and v its unique neighbor in that component. Then v has another neighbor in the component, so $N_H[u] \subsetneq N_H[v]$. Since $(Ax)_w = \sum_{t \in N_H[w]} x_t$ for $w \in S$ (vertices outside S carry zero mass), Lemma 33 gives a contradiction.

($\Leftarrow$) If $\Delta(H) \leq 1$, set $x_i = 1/|S|$ for $i \in S$, $x_i = 0$ otherwise. Every support edge uv is an isolated K_2 in H , so $N_H[u] = N_H[v] = \{u, v\}$ and $(Ax)_u = (Ax)_v = 2/|S|$. $\square$

D.4 Polynomial-time algorithm for forests

Lemma 36 (Closed-form cost for fixed support). *Let $S \subseteq V$ be nonempty. Among all distributions supported within S , the minimum ℓ_1 -cost is*

$$F(S) = \sum_{i \notin S} c_i y_i + (1 - y(S)) \min_{i \in S} c_i.$$

Note: the least-cost distribution above may *not* be an equilibrium.

Proof. Let $m = 1 - y(S)$ and $c_* = \min_{i \in S} c_i$. For any distribution x supported within S , we have $x_i = 0$ for every $i \notin S$, so $\sum_{i \notin S} c_i |x_i - y_i| = \sum_{i \notin S} c_i y_i$. For the in- S part, setting $d_i = x_i - y_i$, we have $\sum_{i \in S} d_i = m$, so $\sum_{i \in S} c_i |d_i| \geq c_* \sum_{i \in S} |d_i| \geq c_* m$.

To attain equality, pick $r \in S$ with $c_r = c_*$ and set $x_i = y_i$ for $i \in S \setminus \{r\}$, $x_r = y_r + m$. $\square$

Theorem 37 (Polynomial-time algorithm for forests). *If G is a forest, the optimization problem is solvable in $O(n^2)$ time. The decision problem on forests is therefore in P .*

Proof. Step 1: reduction to dissociation sets. By Theorem 35, the feasible supports are exactly the nonempty dissociation sets (vertex sets S with $\Delta(G[S]) \leq 1$). Since dissociation sets are hereditary, $\text{OPT} = \min\{F(S) : \emptyset \neq S \subseteq V, \Delta(G[S]) \leq 1\}$.

Step 2: anchor at a minimum-cost vertex. Set $K = \sum_{i \in V} c_i y_i$. For each $r \in V$, let $V_r = \{i \in V : c_i \geq c_r\}$, let $G_r = G[V_r]$, and assign weight $w_i^{(r)} = (c_i + c_r)y_i$ to each $i \in V_r$. Let M_r be the maximum total weight of a dissociation set in G_r containing r .

Using Lemma 36, for any nonempty dissociation set S with anchor $r \in S$ minimizing c_i over S : $F(S) = K + c_r - \sum_{i \in S} w_i^{(r)} \geq K + c_r - M_r$. Conversely, the dissociation set achieving M_r gives $F(S_r) = K + c_r - M_r$. Hence $\text{OPT} = \min_{r \in V} (K + c_r - M_r)$.

Step 3: dynamic programming. Each G_r is a forest. In the component containing r , root the tree at r . For each vertex v in a rooted subtree, define:

- P_v : max weight of a dissociation set in the subtree of v with $v \notin S$,
- Q_v : max weight with $v \in S$ and no child of v in S ,
- R_v : max weight with $v \in S$ and exactly one child of v in S .

Base case (leaf): $P_v = 0$, $Q_v = w(v)$, $R_v = -\infty$. For children $u_1, \dots, u_t$:

$$\begin{aligned} P_v &= \sum_{j=1}^t \max\{P_{u_j}, Q_{u_j}, R_{u_j}\}, \\ Q_v &= w(v) + \sum_{j=1}^t P_{u_j}, \\ R_v &= w(v) + \sum_{j=1}^t P_{u_j} + \max_{1 \leq j \leq t} (Q_{u_j} - P_{u_j}). \end{aligned}$$

(The R_v formula uses the observation that selecting child u_j adds $Q_{u_j} - P_{u_j}$ over the baseline. If $v \in S$ and child $u_j \in S$, then u_j cannot have a selected child, so u_j must be in state Q .)

For the root component, M_r uses $\max\{Q_r, R_r\}$ (the root must be selected). For other components, add $\max\{P_\rho, Q_\rho, R_\rho\}$ at each root ρ . Each M_r is computed in $O(n)$ time; iterating over all anchors takes $O(n^2)$.

Once a minimizing anchor and optimal dissociation set S_r are found, the optimal distribution is $x_i = 0$ for $i \notin S_r$, $x_i = y_i$ for $i \in S_r \setminus \{r\}$, $x_r = y_r + 1 - y(S_r)$. $\square$