TESTING CONTAINMENT OF TROPICAL HYPERSURFACES WITHIN POLYNOMIAL COMPLEXITY

Dima Grigoriev

CNRS, Mathématique, Université de Lille, Villeneuve d'Ascq, 59655, France
e-mail: dmitry.grigoryev@univ-lille.fr
URL: http://en.wikipedia.org/wiki/Dima_Grigoriev

Abstract

For tropical n-variable polynomials f, g a criterion of containment for tropical hypersurfaces $\mathrm{Trop}(f) \subset \mathrm{Trop}(g)$ is provided in terms of their Newton polyhedra $N(f), N(g) \subset \mathbb{R}^{n+1}$. Namely, $\mathrm{Trop}(f) \subset \mathrm{Trop}(g)$ iff for every vertex v of $N(g)$ there exists a unique vertex w of $N(f)$ such that for the tangent cones it holds $v - w + N(f)_w \subseteq N(g)_v$. Relying on this criterion an algorithm is designed which tests whether $\mathrm{Trop}(f) \subset \mathrm{Trop}(g)$ within polynomial complexity.

keywords: containment of tropical hypersurfaces, tangent cones of Newton polyhedra, polynomial complexity algorithm
AMS classification: 14T05

Introduction

Consider a tropical polynomial [6], [8]

$$f = \min_{1 \le i \le k} \{M_i\},\ M_i = \sum_{1 \le j \le n} a_{i,j} x_j + a_{i,0},\ 0 \le a_{i,j} \in \mathbb{Z} \cup \{\infty\},\ a_{i,0} \in \mathbb{R} \cup \{\infty\}. \tag{1}$$

The tropical hypersurface $\mathrm{Trop}(f) \subset \mathbb{R}^n$ consists of points $(x_1, \dots, x_n)$ such that the minimum in (1) is attained at least at two tropical monomials $M_i, 1 \le i \le k$.

For each $1 \le i \le k$ consider the ray $\{(a_{i,1}, \dots, a_{i,n}, a) : a_{i,0} \le a \in \mathbb{R}\} \subset \mathbb{R}^{n+1}$ with the apex at the point $(a_{i,1}, \dots, a_{i,n}, a_{i,0})$. The convex hull of all these rays for $1 \le i \le k$ is Newton polyhedron $N(f)$. Rays of this form we call vertical, and the last coordinate we call vertical. Note that $N(f)$ contains edges of finite length and vertical rays.

A point $(x_1, \dots, x_n) \in \mathrm{Trop}(f)$ iff a parallel shift H'_x of the hyperplane $H_x = \{(z_1, \dots, z_n, x_1z_1 + \cdots + x_nz_n) \ : \ z_1, \dots, z_n \in \mathbb{R}\} \subset \mathbb{R}^{n+1}$ has at least two common points (vertices) with $N(f)$, so that $N(f)$ is located in the half-space above H'_x (with respect to the vertical coordinate). In this case H'_x has (at least) a common edge with $N(f)$, and we say that H'_x supports $N(f)$ at $H'_x \cap N(f)$.

The goal of the paper is to provide for tropical polynomials f, g an explicit criterion of containment $\mathrm{Trop}(f) \subset \mathrm{Trop}(g)$ in terms of Newton polyhedra $N(f), N(g)$. Namely, $\mathrm{Trop}(f) \subset \mathrm{Trop}(g)$ iff for each vertex v of $N(g)$ there exists a unique vertex w of $N(f)$ such that $v-w+N(f)_w \subseteq N(g)_v$ where $N(f)_w$ denotes the tangent cone of $N(f)$ at the vertex w. Relying on this criterion, we design an algorithm which tests whether $\mathrm{Trop}(f) \subset \mathrm{Trop}(g)$ within polynomial bit-complexity.

Note that a criterion of emptiness of a tropical prevariety $\mathrm{Trop}(f_1, \dots, f_l)$ is established in [4] (one can treat this as a tropical weak Nullstellensatz), further developments one can find in [7], [1]. The issue of containment of tropical hypersurfaces is a particular case of an open problem of a tropical strong Nullstellensatz, i.e. a criterion of containment $\mathrm{Trop}(f_1, \dots, f_l) \subseteq \mathrm{Trop}(g)$. We mention that in [5] (which improves [2]) a strong Nullstellensatz is provided for systems of min-plus equations of the form $f = g$ (in terms of congruences of tropical polynomials).

Observe that the family of all tropical prevarieties coincides with the family of all min-plus prevarieties (and both coincide with the family of all finite unions of polyhedra given by linear constraints with rational coefficients [8]). On the other hand, the issue of a strong Nullstellensatz is different for these two types of equations.

1 Containment of tropical hypersurfaces and tangent cones

Theorem 1.1 *Assume that each of tropical polynomials f, g has at most of k tropical monomials of the form $i_1x_1 + \cdots + i_nx_n + a$ where integers $|a|, i_1, \dots, i_n \le 2^d$. There is an algorithm which tests whether* $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$ *within bit-complexity $O((n+k)^{1.5}nk^3d)$. In particular, the bit-complexity is polynomial in the bit-size of the input $2k(n+1)d$.*

The proof of the theorem relies on the following criterion of containment of tropical hypersurfaces.

Proposition 1.2 *For tropical polynomials f, g in n variables it holds* $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$ *iff for each vertex v the Newton polyhedron $N(g) \subset \mathbb{R}^{n+1}$ there exists a vertex w of $N(f)$ such that $v - w + N(f)_w \subseteq N(g)_v$. Moreover, for each vertex v the vertex w is unique, and for any hyperplane $H \subset \mathbb{R}^{n+1}$ such that $H \cap N(g)_v = \{v\}$ there exists a (unique) hypeplane H_0 parallel to H for which it holds $H_0 \cap N(f)_w = \{w\}$.*

Proof of the proposition. First assume that for each vertex v of $N(g)$ there exists a vertex w of $N(f)$ such that $v - w + N(f)_w \subseteq N(g)_v$. Suppose that $\mathrm{Trop}(f) \not\subseteq \mathrm{Trop}(g)$, then there exists a hyperplane $\mathbb{R}^{n+1} \supset H \in \mathrm{Trop}(f) \setminus \mathrm{Trop}(g)$ (cf. the description of a tropical hypersurface as a set of hyperplanes in the introduction). Therefore, a parallel shift H_0 of H supports $N(g)$ at its single vertex v. By the assumption, $v - w + N(f)_w \subseteq N(g)_x$ for some vertex w of $N(f)$. Hence the hyperplane $w - v + H_0$ supports $N(f)$ at its single vertex w. This leads to a contradiction with that $H \in \mathrm{Trop}(f)$. Thus, $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$.

Conversely, assume that $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$. For a vertex v of $N(g)$ choose a supporting hyperplane H at v such that $H \cap N(g)_v = \{v\}$. Then there exists a unique vertex w of $N(f)$ such that $(w - v + H) \cap N(f) = \{w\}$ taking into account that $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$.

We claim that $v - w + N(f)_w \subseteq N(g)_v$. Suppose the contrary. Then there exists a ray $L \subset v - w + N(f)_w$ such that $L \cap N(g)_v = \{v\}$. Therefore, there exists a hyperplane H_1 such that $L \subset H_1,\ H_1 \cap N(g)_v = \{v\}$. Hence $H_1 \in \mathrm{Trop}(f) \setminus \mathrm{Trop}(g)$. The obtained contradiction proves the claim.

Finally, we justify the uniqueness of a vertex w in the proposition, in other words, its independence of a choice of a hyperplane H supporting $N(g)$ at v. Suppose the contrary. Then there exists a supporting hyperplane H which supports $N(f)$ at least at two vertices, since the space of supporting hyperplanes (such that $H \cap N(g)_v = \{v\}$) is connected. This contradicts to $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$. $\square$

In other words, Proposition 1.2 means that $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$ iff each cone of the highest dimension of the normal fan of $N(g)$ is contained in a cone (of the highest dimension) of the normal fan of $N(f)$. We mention that it is known that $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$ iff it holds for the Minkowski sum $t \cdot N(f) + P = N(g)$ for suitable $t > 0$ and a polyhedron P (however, an exact reference is unknown to the author). This result implies Proposition 1.2 in one direction: namely, that the containment $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$ entails the criterion of containment in Proposition 1.2. On the other hand, the criterion of containment from Proposition 1.2 is more relevant to design an algorithm to test whether $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$.

In the construction of the algorithm asserted in Theorem 1.1 we stick with straight-line program [3] as a computational model.

Proof of the Theorem. We repeatedly involve a linear programming algorithm which has bit-complexity $O((n+m)^{1.5}nL)$ [10] where n denotes the number of variables, m denotes the number of (linear) constraints, and L denotes the bit-size of the input. Moreover, the algorithm produces a solution from $\mathbb{Q}^n$ of a linear programming problem (provided that it does exist) with bit-size $O(n^2L)$.

Therefore for given points $v, v_1, \dots, v_m \in \mathbb{Z}^n$ with absolute values of their coordinates at most 2^d one can test whether v belongs to the convex hull $conv\{v_1, \dots, v_m\}$ within bit-complexity $O((n+m)^{1.5}nm^2d)$. Indeed, one has to solve a linear programming problem:

$$v = b_1v_1 + \cdots + b_mv_m,\ 0 \leq b_1, \dots, b_m \leq 1.$$

Applying the latter subroutine to the problems whether v_i belongs to $conv\{v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_m\}$, $1 \leq i \leq m$, one can find the vertices of the polytope $conv\{v_1, \dots, v_m\}$ within bit-complexity $O((n+m)^{1.5}nm^3d)$.

We agree that a tropical monomial $i_1x_1 + \cdots + i_nx_n + a$ corresponds to the point $(i_1, \dots, i_n, a) \in \mathbb{Z}^{n+1}$. One can find the vertices of Newton polyhedron $N(g)$ within bit-complexity $O((n+k)^{1.5}nk^3d)$ (as well as the vertices of $N(f)$) as follows. Let $v_1, \dots, v_k \in \mathbb{Z}^{n+1}$ be the points corresponding to the tropical monomials of g. Then $v_i, 1 \leq i \leq k$ is a vertex of $N(g)$ iff the following linear programming problem has no solution:

$$v_i - b(0, \dots, 0, 1) = \sum_{1 \leq j \neq i \leq k} b_jv_j,\ b \geq 0, 0 \leq b_j \leq 1, 0 \leq j \neq i \leq k. \tag{2}$$

W.l.o.g. assume that the vertices of $N(g)$ are $v_1, \dots, v_s$, $s \leq k$ (they are just the vertices on the bottom of the polytope $conv\{v_1, \dots, v_k\}$). Similarly, assume that $w_1, \dots, w_t$, $t \leq k$ are the vertices of $N(f)$.

For each vertex v_i, $1 \leq i \leq s$ the algorithm produces a vector $u_i \in \mathbb{Z}^{n+1}$ solving the following linear programming problem:

$$\langle u_i, v_j - v_i \rangle > 0,\ 1 \leq j \neq i \leq s,\ \langle u_i, (0, \dots, 0, 1) \rangle > 0. \tag{3}$$

Then the hyperplane $H_i \subset \mathbb{R}^{n+1}$ orthogonal to u_i supports $N(g)$ at its single vertex v_i. The bit-complexity of producing u_i does not exceed $O((n+k)^{1.5}nkd)$. The bit-size of u_i is bounded by $O(n^2d)$. Denote $v_0 := v_i + (0, \dots, 0, 1)$.

After that the algorithm finds a vertex w_q of the polyhedron $N(f)$ with the minimal value of $\langle w_q, u_i \rangle$. This can be executed within bit-complexity $O(n^2kd)$. If a vertex w_q is not unique then the hyperplane parallel to H_i and passing through w_q does not support $N(f)$ at its single vertex w_q, therefore $H_i \in \mathrm{Trop}(f) \setminus \mathrm{Trop}(g)$, and in this case the algorithm outputs that $\mathrm{Trop}(f) \nsubseteq \mathrm{Trop}(g)$ and halts. Denote $w_l' := v_i + w_l - w_q, 1 \leq l \leq t$.

Finally, for each $1 \le l \ne q \le t$ the algorithm solves the following linear programming problem:

$$\langle u, w_l' - v_i \rangle = 0,\ \langle u, v_j - v_i \rangle > 0, 0 \le j \ne i \le s,\ \langle u, w_p' - v_i \rangle \ge 0, 1 \le p \le t. \quad (4)$$

If (4) has a solution $u \in \mathbb{Q}^{n+1}$ then the hyperplane orthogonal to u belongs to $\mathrm{Trop}(f) \setminus \mathrm{Trop}(g)$. In this case the algorithm outputs that $\mathrm{Trop}(f) \not\subseteq \mathrm{Trop}(g)$ and halts. Otherwise, if (4) has no solutions for each $1 \le i \le s,\ 1 \le l \ne q \le t$ then the algorithm outputs that $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$. The bit-complexity of solving the systems (4) can be bounded by $O((n+k)^{1.5}nk^2d)$.

The correctness of the algorithm follows from Proposition 1.2. $\square$

Now we summarize the algorithm testing whether $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$ designed in the proof of Theorem 1.1.

• The algorithm finds the vertices $v_1, \dots, v_s$ of $N(g)$: namely, v_i is a vertex of $N(g)$ iff (2) has no solutions. Similarly, the algorithm finds the vertices $w_1, \dots, w_t$ of $N(f)$.

• For each $1 \le i \le s$ the algorithm produces a vector u_i satisfying (3). Denote by H_i the hyperplane orthogonal to u_i. The algorithm finds a vertex w_q of $N(f)$ with the minimal value of $\langle w_q, u_i \rangle$. If the vertex w_q is not unique then the hyperplane $H_i \in \mathrm{Trop}(f) \setminus \mathrm{Trop}(g)$, and the algorithm halts.

• For each point $w_l' := v_i + w_l - w_q, 1 \le l \ne q \le t$ the algorithm tests whether (4) has a solution u. If it is the case then the hyperplane orthogonal to u belongs to $\mathrm{Trop}(f) \setminus \mathrm{Trop}(g)$, and the algorithm halts. Otherwise, if (4) has no solutions for $1 \le i \le s,\ 1 \le l \ne q \le t$ then the algorithm outputs that $\mathrm{Trop}(f) \subseteq \mathrm{Trop}(g)$.

It would be interesting to provide a criterion of containment for tropical prevarieties $\mathrm{Trop}(f_1, \dots, f_k) \subseteq \mathrm{Trop}(g)$. Note that the latter problem is NP-hard [9].

Acknowledgements. The author is grateful to anonymous referees for valuable remarks.

References

[1] M. Akian, A. Béreau and S. Gaubert. The tropical Nullstellensatz and Positivstellensatz for sparse polynomial systems. *ACM Proc. Int. Symp. Symb. Alg. Comput.*, 43-52, 2023.

[2] A. Bertram and R. Easton. The tropical Nullstellensatz for congruences. *Adv. Math.*, 308:36-82, 2017.

[3] P. Bürgisser, M. Clausen and A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften.* Springer, 1997.

[4] D. Grigoriev and V. Podolskii. Tropical effective primary and dual Nullstellensaetze. *Discr. Comput. Geometry*, 59:507–552, 2018.

[5] D. Joo and K. Mincheva. Prime congruences of additively idempotent semirings and a Nullstellensatz for tropical polynomials. *Selecta Math.*, 24:2207-2233, 2018.

[6] M. Joswig. *Essentials of Tropical Combinatorics*, volume 219 of *Graduate Studies in Mathematics.* American Mathematical Society, 2021.

[7] D. Maclagan and F. Rincon. Tropical ideals. *Compos. Math.*, 154:640-670, 2018.

[8] D. Maclagan and B. Sturmfels. *Introduction to Tropical Geometry*, volume 161 of *Graduate Studies in Mathematics.* American Mathematical Society, 2015.

[9] T. Theobald. On the frontiers of polynomial computations in tropical geometry. *J. Symb. Comput.*, 41:1360-1375, 2006.

[10] P. Vaidya. Speeding-up linear programming using fast matrix multiplication. *Proc. IEEE Symp. Found. Comput. Sci.*, 332-337, 1989.