Computational Aspects of the Colorful Carathéodory Theorem

Wolfgang Mulzer[*] *Yannik Stein*[†]

Institut für Informatik, Freie Universität Berlin, Germany
{mulzer, yannikstein}@inf.fu-berlin.de

Abstract

Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be d-dimensional point sets such that the convex hull of each P_i contains the origin. We call the sets P_i *color classes*, and we think of the points in P_i as having color i. A *colorful choice* is a set with at most one point of each color. The *colorful Carathéodory theorem* guarantees the existence of a colorful choice whose convex hull contains the origin. So far, the computational complexity of finding such a colorful choice is unknown.

We approach this problem from two directions. First, we consider approximation algorithms: an *m-colorful choice* is a set that contains at most m points from each color class. We show that for any constant $\varepsilon > 0$, an $\lceil \varepsilon(d+1) \rceil$-colorful choice containing the origin in its convex hull can be found in polynomial time. This notion of approximation has not been studied before, and it is motivated through the applications of the colorful Carathéodory theorem in the literature. In the second part, we present a natural generalization of the colorful Carathéodory problem: in the *Nearest Colorful Polytope* problem (NCP), we are given sets $P_1, \dots, P_n \subset \mathbb{R}^d$ that do not necessarily contain the origin in their convex hulls. The goal is to find a colorful choice whose convex hull minimizes the distance to the origin. We show that computing local optima for the NCP problem is PLS-complete, while computing a global optimum is NP-hard.

1 Introduction

Let $P \subset \mathbb{R}^d$ be a point set. Carathéodory's theorem [5, Theorem 1.2.3] states that if $\vec{0} \in \text{conv}(P)$, there is a subset $P' \subseteq P$ of at most $d+1$ points with $\vec{0} \in \text{conv}(P')$. Bárány [2] gives a generalization to the colorful setting:

Theorem 1.1 (Colorful Carathéodory Theorem [2])**.** *Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be point sets (the color classes). If $\vec{0} \in \text{conv}(P_i)$, for $i = 1, \dots, d+1$, there is a* colorful choice *C with $\vec{0} \in \text{conv}(C)$. Here, a colorful choice is a set with at most one point from each color class.*

Proof sketch. Let C be some colorful choice. If $\vec{0} \in \text{conv}(C)$, we are done. Otherwise, let x be the point on $\text{conv}(C)$ closest to the origin and let h be the hyperplane through x normal to the segment $\vec{0}x$. Since x is a convex combination of at most d points from C, there is a color class P_i that does not contribute to x. Let $p \in C$ be the point of color i in the colorful choice. As $\vec{0} \in \text{conv}(P_i)$, there is a point $p' \in P_i$ that is separated from $\text{conv}(C)$ by h, and $\text{conv}(C \setminus \{p\} \cup \{p'\})$

[*] Supported in part by DFG Grants MU 3501/1 and MU 3501/2.

[†] Supported by the Deutsche Forschungsgemeinschaft within the research training group 'Methods for Discrete Structures' (GRK 1408).

is strictly closer to the origin. There are only finitely many colorful choices, so eventually we must have $\vec{0} \in \operatorname{conv}(C)$. ■

Theorem 1.1 implies Carathéodory's theorem by setting $P_1 = \cdots = P_{d+1}$. Moreover, there are many variants with weaker assumptions [6]. While Carathéodory's theorem can be cast as a linear system and thus be implemented in polynomial time, very little is known about the algorithmic complexity of the colorful Carathéodory theorem [3]. This question is particularly interesting because Sarkaria's proof [11] of Tverberg's theorem [13] gives a polynomial-time reduction from computing Tverberg partitions to computing a colorful choice with the origin in its convex hull. Both problems lie in *Total Function NP* (TFNP), the complexity class of total search problems that can be solved in non-deterministic polynomial time. It is well known that no problem in TFNP is NP-hard unless NP = coNP [4]. Recently, Meunier and Sarrabezolles [7] have shown that a related problem is complete for a subclasses of TFNP: given $d+1$ pairs of points $P_1, \ldots, P_{d+1} \in \mathcal{Q}^d$ and a colorful choice that contains the origin in its convex hull, it is PPAD-complete [10] to find another colorful choice that contains the origin in its convex hull.

Since we have no exact polynomial-time algorithms for the colorful Carathéodory theorem, approximation algorithms are of interest. This was first considered by Bárány and Onn [3] who described how to find a colorful choice whose convex hull is "close" to the origin. Let $\varepsilon, \rho > 0$ be parameters. We call a set *ε-close* if its convex hull has distance at most ε to the origin. Given sets $P_1, \ldots, P_{d+1} \in \mathcal{Q}^d$ s.t. (i) each P_i contains a ball of radius ρ centered at the origin in its convex hull, (ii) all points $p \in P_i$ fulfill $1 \leq \|p\| \leq 2$, and (iii) the points in all sets can be encoded using L bits, one can find a colorful choice C that is ε-close to the origin in time $\operatorname{poly}(L, \log(1/\varepsilon), 1/\rho)$ on the Word-Ram with logarithmic costs. If $1/\rho = O(\operatorname{poly}(L))$, the algorithm actually finds a colorful choice with the origin in its convex hull.

However, when using the colorful Carathéodory theorem in the proof of another statement, it is often crucial that the convex hull of the colorful choice contains the origin. Being "close" is not enough. On the other hand, allowing multiple points from each color class may have a natural interpretation in the reduction. For example, this is the case in Sarkaria's proof [11] of Tverberg's theorem and in the proof of the First Selection Lemma [5, Theorem 9.1.1]. This motivates a different notion of approximation: we need a "colorful" set with the origin in its convex hull, but we may take more than one point from each color. More formally, given a parameter m and sets $P_1, \ldots, P_{d+1} \in \mathcal{Q}^d$, find a set C s.t. $\vec{0} \in \operatorname{conv}(C)$ and s.t. for all P_i, we have $|C \cap P_i| \leq m$. In contrast to the setting considered by Bárány and Onn, we have no general position assumption. Surprisingly, this notion does not seem to have been studied before.

Coming from another direction, as a first step towards understanding what makes the problem hard, we consider the *Nearest Colorful Polytope (NCP) problem*, a natural generalization inspired by the proof of Theorem 1.1. Given color classes $P_1, \ldots, P_n \subset \mathbb{R}^d$, not necessarily containing the origin in their convex hulls, find a colorful choice whose convex hull minimizes the distance to the origin. We study two variants: the local search problem, where we want to find a colorful choice whose convex hull cannot be brought closer to the origin by exchanging a *single* point with another point of the same color; and the global search problem, where we want to compute a colorful choice with minimum distance to the origin. We refer to these problems as L-NCP and G-NCP, respectively. L-NCP is particularly interesting since Bárány's proof of the colorful Carathéodory theorem gives a local search algorithm. The complexity of G-NCP was posed as an open problem by Bárány and Onn [3]. This question was also answered independently by Meunier and Sarrabezolles [7].

1.1 Our Results

Given sets $P_1, \dots, P_n \subset \mathbb{R}^d$, we call a set C containing at most m points from each set P_i an *m-colorful choice.* A 1-colorful choice is also called *perfect colorful choice.* All presented algorithms are analyzed on the Real-Ram model with unit costs. We begin with an approximation algorithm based on a simple dimension reduction argument. This leads to the following result:

Proposition 1.2. *Let $P_1, \dots, P_{\lfloor d/2 \rfloor +1} \subset \mathbb{R}^d$ be $\lfloor d/2 \rfloor + 1$ sets of size at most $d+1$ that each contain the origin in their convex hulls. Then, a $(\lceil d/2 \rceil + 1)$-colorful choice containing the origin in its convex hull can be computed in $O(d^5)$ time.*

Generalizing the algorithm from Proposition 1.2, we can further improve the approximation guarantee by repeatedly combining approximations for lower dimensional linear subspaces.

Theorem 1.3. *Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be sets of size at most $d+1$ s.t. $\vec{0} \in \operatorname{conv}(P_i)$ for all $i = 1, \dots, d+1$. Then, for any $\varepsilon = \Omega(d^{-1/3})$, an $\lceil \varepsilon(d+1) \rceil$-colorful choice containing the origin in its convex hull can be computed in $d^{O((1/\varepsilon)\log(1/\varepsilon))}$ time.*

In particular, for any constant ε the algorithm from Theorem 1.3 runs in polynomial-time. Given $\Theta(d^2 \log d)$ color classes, we can also improves the naive $d^{O(d)}$ algorithm asymptotically:

Theorem 1.4. *Let $P_1, \dots, P_n \subset \mathbb{R}^d$ be $n = \Theta(d^2 \log d)$ sets of size at most $d+1$ s.t. $\vec{0} \in \operatorname{conv}(P_i)$, for $i = 1, \dots, n$. Then, a perfect colorful choice can be computed in $d^{O(\log d)}$ time.*

On the other hand, if we are given only two color classes, we can achieve a $d - \Theta(\sqrt{d})$ approximation guarantee:

Proposition 1.5. *Let $P, Q \subset \mathbb{R}^d$ be two sets of size at most $d+1$ that both contain the origin in their convex hulls. Then, a $(d - \Theta(\sqrt{d}))$-colorful choice can be computed in $O(d^4)$ time.*

On the hardness side, we show that a generalization of the colorful Carathéodory problem, the *local search nearest colorful polytope (L-NCP)* problem, is complete for the complexity class *polynomial-time local search* (PLS). PLS contains local-search problems for which a single improvement step can be carried out in polynomial-time, but the total length of the search path may be exponential. Using essentially the same reduction, we can also prove that finding a global optimum for NCP (G-NCP) is NP-hard.

Theorem 1.6. *L-NCP is PLS-complete.*

Theorem 1.7. *G-NCP is NP-hard.*

2 Approximating the Colorful Carathéodory Theorem

Throughout the paper, we denote for a given point set $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^d$ by

- $\operatorname{span}(P) = \{\sum_{i=1}^n \alpha_i p_i \mid \alpha_i \in \mathbb{R}\}$ its linear span and by $\operatorname{span}(P)^\perp = \{v \in \mathbb{R}^d \mid \forall p \in \operatorname{span}(P) : \langle v, p \rangle = 0\}$ the subspace orthogonal to $\operatorname{span}(P)$;
- $\operatorname{aff}(P) = \{\sum_{i=1}^n \alpha_i p_i \mid \alpha_i \in \mathbb{R}, \sum_{i=1}^n \alpha_i = 1\}$ its affine hull;
- $\operatorname{pos}(P) = \{\sum_{i=1}^n \mu_i p_i \mid \mu_i \geq 0\}$ all linear combinations with nonnegative coefficients;
- $\operatorname{conv}(P) = \{\sum_{i=1}^n \lambda_i p_i \mid \lambda_i \geq 0, \sum_{i=1}^n \lambda_i = 1\}$ its convex hull; and by
- $\dim(P)$ the dimension of $\operatorname{span}(P)$.

Furthermore, we say that a set $P \subset \mathbb{R}^d$ is in *general position* if for every $k \leq d$, no $k+2$ points lie in a k-flat *and* if no proper subset of P contains the origin in its convex hull. We also use the following constructive version of Carathéodory's theorem:

Lemma 2.1. *Let $P \subset \mathbb{R}^d$ be a set of $O(d)$ points that contains the origin in its convex hull. In $O(d^4)$ time, we can find a subset $P' \subseteq P$ of at most $d+1$ points in general position such that P' contains the origin in its convex hull.*

2.1 Simple Approximations

Since there are no known approximation algorithms for computing m-colorful choices, even simple ones are of interest to gain some intuition for the problem. It is a straightforward exercise to show that a $(d-1)$-colorful choice can be computed in polynomial-time. However, even $m = d-2$ seems to be nontrivial.

In this section, we present two algorithms that both compute a $(d+1)/2$-colorful choice in $O(d^5)$ time, but differ in the number of required color classes. The following lemma is the key ingredient of both algorithms: it enables us to replace each color class P_i by two points v_1, v_2, so that each point represents half of the points in P_i. We call the points v_1, v_2 *representatives* for P_i. Now, a perfect colorful choice for the representatives will correspond to a $\lceil (d+1)/2 \rceil$-colorful choice for the original points. The presented algorithms differ only in the way the perfect colorful choice is computed for this special case of the colorful Carathéodory problem. The first one uses basic linear algebra, while the second one is based on a simple dimension reduction argument.

Lemma 2.2. *Let $P \subset \mathbb{R}^d$, $2 \leq |P| \leq d+1$, be a set in general position that contains the origin in its convex hull. Then, for every partition of P into two sets P_1, P_2, there is a vector $v \neq \vec{0}$ s.t. $v \in \operatorname{pos}(P_1)$ and $-v \in \operatorname{pos}(P_2)$. This vector can be found in $O(d^3)$ time.*

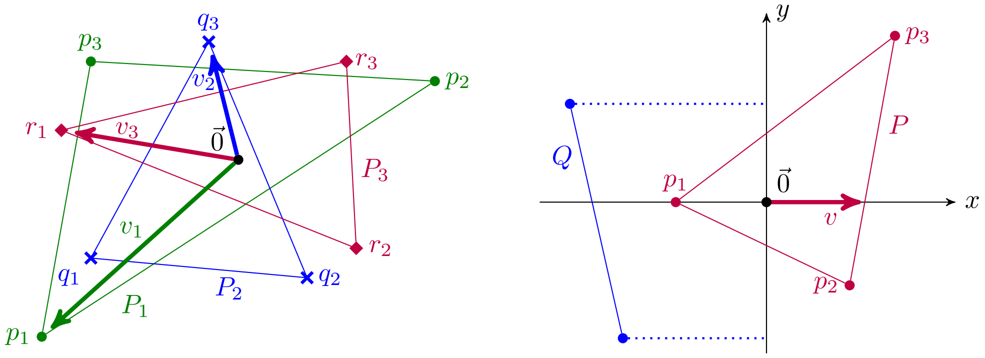
Proof. Write $\vec{0}$ as $\vec{0} = \sum_{p \in P} \lambda_p p$, such that $\lambda_p \geq 0$ for all $p \in P$ and such that $\sum_{p \in P} \lambda_p = 1$. The coefficients λ_p can be computed in $O(d^3)$ time. Since P is in general position, we have $\lambda_p > 0$ for all $p \in P$. Set $v = \sum_{p \in P_1} \lambda_p p$. By construction, we have $v \neq \vec{0}$, $v \in \operatorname{pos}(P_1)$, and $-v \in \operatorname{pos}(P_2)$. ■

In the first algorithm, we partition each set P_i into two sets of equal size and apply Lemma 2.2 to obtain $d+1$ representatives $v_1, \ldots, v_{d+1}$. The set $\{v_1, \ldots, v_{d+1}\}$ must be linearly dependent, so we can use a nontrivial $\vec{0}$-combination to find a perfect colorful choice for the representatives.

Proposition 2.3. *Let $P_1, \ldots, P_{d+1} \subset \mathbb{R}^d$ be $d+1$ sets s.t. $|P_i| \leq d+1$ and s.t. P_i contains the origin in its convex hull, for $i = 1, \ldots, d+1$. Then, a $\lceil (d+1)/2 \rceil$-colorful choice can be computed in $O(d^5)$ time.*

Proof. First, prune each set P_i, $i = 1, \ldots, d+1$, with Lemma 2.1. This requires $O(d^5)$ time. Assume w.l.o.g. that all sets still contain at least two points (since otherwise at least one set contains the origin). Partition each set P_i arbitrarily into two sets $P_{i,1}, P_{i,2}$ of equal size and let $v_1, \ldots, v_{d+1}$ be the vectors obtained by applying Lemma 2.2 to the partitions. Since these vectors are linearly dependent, there is a nontrivial linear combination of $\vec{0}$: $\vec{0} = \sum_{i=1}^{d+1} \mu_i v_i$. The coefficients μ_i can be computed in $O(d^3)$ time by solving a linear equation system. For each vector v_i with $\mu_i > 0$, take $P_{i,1}$ (since $v_i \in \operatorname{pos}(P_{i,1})$), otherwise $P_{i,2}$ (since $-v_i \in \operatorname{pos}(P_{i,2})$). Figure 1a shows an example in two dimensions. The overall running time is dominated by the initial pruning step. ■

Lemma 2.2 can also be used to reduce the dimension by one. We repeat this until the dimension is small enough, i.e., $\lceil d/2 \rceil$, and then simply apply Lemma 2.1 in the low dimensional



(a)

(b)

Fig. 1: (a) Example of Proposition 2.3 in two dimensions. The color classes are partitioned into $P_1 = \{p_1\}\dot\cup\{p_2, p_3\}$, $P_2 = \{q_3\}\dot\cup\{q_1, q_2\}$, and $P_3 = \{r_1\}\dot\cup\{r_2, r_3\}$. The set $C = \{p_1\}\dot\cup\{q_3\}\dot\cup\{r_2, r_3\}$ is a 2-colorful choice. (b) Example of Proposition 1.2 in two dimensions. The representative v is computed for the partition $P = \{p_2, p_3\}\dot\cup\{p_1\}$. W.l.o.g. assume v lies on the x-axis. The set Q is a recursively computed approximation that contains the origin in its convex hull if projected onto the y-axis. The set $C = Q \cup \{p_2, p_3\}$ is a 2-colorful choice containing the origin in its convex hull.

space. This algorithm requires only $\lfloor d/2 \rfloor + 1$ color classes instead of $d + 1$. We will generalize it in the next section.

Proof of Proposition 1.2. We prune P_1 with Lemma 2.1. If $|P_1| = 1$, we have $P_1 = \{\vec{0}\}$, and P_1 is a valid approximation. If $|P_1| \geq 2$, we partition P_1 arbitrarily into two sets $P_{1,1}, P_{1,2}$ of equal size. We apply Lemma 2.2 to obtain a vector v. We project the remaining color classes onto the orthogonal subspace $\text{span}(v)^{\perp}$ and recursively compute a $(\lceil d/2 \rceil + 1)$-colorful choice $\widetilde{C}$ for the projection. Let C' be the d-dimensional point set corresponding to $\widetilde{C}$. If the convex hull of C' intersects $\text{pos}(v)$, we set $C = C' \cup P_{1,2}$ (since $-v \in \text{pos}(P_{i,2})$), otherwise, we set $C = C' \cup P_{i,1}$ (since $v \in \text{pos}(P_{i,1})$). In both cases, C is a $(\lceil d/2 \rceil + 1)$-colorful choice with the origin in its convex hull. See Figure 1b. If only one color is left, i.e., if we are in dimension $d - \lfloor d/2 \rfloor = \lceil d/2 \rceil$, we prune this color with Lemma 2.1 and we return the resulting set of size at most $\lceil d/2 \rceil + 1$.

Each invocation of Lemma 2.1 and of Lemma 2.2 takes $O(d^4)$ time. The recursion depth is bounded by $\lfloor d/2 \rfloor + 1$, which results in a total running time of $O(d^5)$, as claimed. ■

2.2 Approximation by Rebalancing

The algorithm from Proposition 1.2 prunes half of the points from each color class in a complete run. We generalize this approach in two respects: first, we repeatedly prune points to improve the approximation guarantee. Second, we reduce the dimensionality in each step by more than one to improve the running time.

Let $P_1, \ldots, P_{d+1} \subset \mathbb{R}^d$ be the color classes and $\lceil \varepsilon(d + 1) \rceil$ be the desired approximation guarantee. Throughout the execution of the algorithm, we maintain a temporary approximation $C \subset P_1 \cup \cdots \cup P_{d+1}$ that contains the origin in its convex hull, but may have more than $\lceil \varepsilon(d+1) \rceil$ points of the same color. Initially, C is a complete color class. Using the following lemma, we can replace a single point in C by an approximate colorful choice for the orthogonal space $\text{span}(C)^{\perp}$.

Lemma 2.4. *Let $C \subset \mathbb{R}^d$, $|C| = k \leq d+1$, be a set in general position that contains the origin in its convex hull. Furthermore, let $Q \subset \mathbb{R}^d$ be a set of size $O(d)$ whose orthogonal projection onto* $\operatorname{span}(C)^\perp$ *contains the origin in its convex hull. Then, there is a point $c \in C$ computable in $O(d^4)$ time s.t.* $\vec{0} \in \operatorname{conv}(Q \cup C \setminus \{c\})$.

Proof. Write Q as $Q = \{q_1, \dots, q_l\}$. Each q_i can be expressed as $\widetilde{q}_i + \hat{c}_i$, where $\widetilde{q}_i$ denotes the orthogonal projection of q_i onto $\operatorname{span}(C)^\perp$ and $\hat{c}_i \in \operatorname{span}(C)$. By our assumption, the origin is a convex combination of $\widetilde{q}_1, \dots, \widetilde{q}_l$: $\vec{0} = \sum_{i=1}^l \lambda_i \widetilde{q}_i$, where $\lambda_i \geq 0$ and $\sum_{i=1}^l \lambda_i = 1$. Consider the convex combination $q = \sum_{i=1}^l \lambda_i q_i$ of points in Q with the same coefficients. Since $q = \sum_{i=1}^l \lambda_i q_i = \sum_{i=1}^l \lambda_i(\widetilde{q}_i + \hat{c}_i) = \sum_{i=1}^l \lambda_i \hat{c}_i$, q is contained in $\operatorname{span}(C)$.

By our assumption, we have $\vec{0} \in \operatorname{conv}(C)$. Since C is in general position, the following lemma implies that $\operatorname{pos}(C) = \operatorname{span}(C)$:

Lemma 2.5. *Let $C \subset \mathbb{R}^d$ be a set in general position. Then,* $\vec{0} \in \operatorname{conv}(C)$ *if and only if* $\operatorname{span}(C) \subseteq \operatorname{pos}(C)$.

Proof. "$\Rightarrow$" Write $\vec{0}$ as $\vec{0} = \sum_{c_i \in C} \lambda_i c_i$, where $\sum_{c_i \in C} \lambda_i = 1$ and all $\lambda_i > 0$. The last part holds due to general position. Thus, for all $c_i \in C$, the point $-c_i$ can be expressed as a convex combination of $C \backslash \{c_i\}$. Hence, $\operatorname{span}(c_i)$ is contained in $\operatorname{pos}(C)$ for all i and thus $\operatorname{span}(C) \subseteq \operatorname{pos}(C)$. "$\Leftarrow$" By the assumption, the origin can be expressed as a positive combination of the points in C. Scaling the coefficients so that they sum up to 1 concludes the proof. ■

Thus, there are $k-1$ points $c_{j_1}, \dots, c_{j_{k-1}}$ in C s.t. $-q \in \operatorname{pos}(c_{j_1}, \dots, c_{j_{k-1}})$. We can take $c \in C$ as the single point that does not appear in $c_{j_1}, \dots, c_{j_{k-1}}$.

This point can be found in $O(d^4)$ time by solving $k \leq d+1$ linear equation systems $L_1, \dots, L_k$, where L_j is defined as $\sum_{c_i \in C, i \neq j} \alpha_i c_i = -q$. Since C is in general position, all $(k-1)$-subsets of C are a basis for $\operatorname{span}(C)$. Thus, the linear systems have unique solutions. Furthermore, because C contains the origin in its convex hull, one of the linear systems has a solution with no negative coefficients. ■

Unfortunately, we cannot control which point is replaced when applying Lemma 2.4. We always want to replace a point whose color appears more than $\lceil \varepsilon(d+1) \rceil$ times in C to reduce the maximum number of points that C contains from one color. Generalizing Lemma 2.2, the following lemma enables us to compute representatives for partitions of arbitrary size. Instead of applying Lemma 2.4 to C, we replace one of the representatives for C. By choosing the partition for the representatives appropriately, we can influence the color of the removed points.

Lemma 2.6. *Let $C \subset \mathbb{R}^d$, $|C| \leq d+1$, be a set in general position that contains the origin in its convex hull and let $C_1, \dots, C_m$ be a partition of C. Then, we can find in $O(d^3)$ time a set $C' = \{c'_1, \dots, c'_m\} \subset \mathbb{R}^d$ with the following properties:*

1. $\forall i = 1, \dots, m$: $c'_i \in \operatorname{pos}(C_i) \setminus \{\vec{0}\}$
2. $\vec{0} \in \operatorname{conv}(C')$
3. $\dim(C') = m - 1$

We call the points in C' representatives *for C w.r.t. the partition $C_1, \dots, C_m$.*

Proof. Since C contains the origin in its convex hull, we can write $\vec{0}$ as $\vec{0} = \sum_{c \in C} \lambda_c c$, where all $\lambda_c > 0$, since C is in general position. Define c'_j as $c'_j = \sum_{c \in C_j} \lambda_c c$ for all $i = 1, \dots, m$. Properties 1. and 2. can be easily verified for the set $C' = \{c'_1, \dots, c'_m\}$. Furthermore, c'_1 can be expressed as a linear combination of the other points in C': $c'_1 = -(c'_2 + \dots + c'_m)$. Thus, $\dim(C') < m$. On the other hand, we have $\dim(C') \geq m-1$ due to general position. This proves Property 3. ■

Now, we are ready to put everything together. The algorithm repeatedly replaces points in C by a recursively computed approximate colorful choice for a linear subspace. We are given as input the color classes $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$, each containing the origin in its convex hull, a base case threshold $d_0 \in \mathbb{N}$ and two parameter functions $\mathcal{M} : \mathbb{N}_0 \to \mathbb{N}$ and $\mathcal{D} : \mathbb{N}_0 \to \mathbb{N}$. The first function returns for a given recursion depth the desired approximation guarantee. After completion, the algorithm outputs an $\mathcal{M}(0)$-colorful choice. The second function, $\mathcal{D} : \mathbb{N}_0 \to \mathbb{N}$, controls the dimension reduction. It returns for a given recursion depth j the desired dimension of the problem. We require that the parameter functions fulfill the following conditions:

1. $\mathcal{M}$ and $\mathcal{D}$ are strictly decreasing and can be computed in $O(d^4)$ time;
2. $\mathcal{D}(0) = d$; and
3. for all $\mathcal{D}(j) > d_0$, the following inequalities hold

$$\left\lfloor \frac{\mathcal{D}(j) + 1}{\mathcal{M}(j) - \mathcal{M}(j+1)} \right\rfloor \leq \mathcal{D}(j) - \mathcal{D}(j+1) \leq \mathcal{M}(j).$$

We call a pair of parameter functions *feasible* if the above requirements hold for this pair.

Suppose we are at recursion depth j. That is, the input points are $\mathcal{D}(j)$ dimensional and we want to compute an $\mathcal{M}(j)$-colorful choice. If $\mathcal{D}(j)$ is less than our base case threshold d_0, we compute an approximation by brute force. Otherwise, we initialize the temporary approximation C with a complete color class and prune it with Lemma 2.1. As long as C is not an $\mathcal{M}(j)$-colorful choice, we repeat the following steps: we partition C into $k = \mathcal{D}(j) - \mathcal{D}(j+1) + 1$ sets $C_1, \dots, C_k$, where the points from each color in C are distributed evenly among the k sets. Let $n_i = |P_i \cap C|$ denote the number of points from P_i in C. Since $k \leq \mathcal{M}(j) + 1$, each set in the partition contains at least one point from each color class P_i for which $n_i \geq \mathcal{M}(j) + 1$. Applying Lemma 2.6, we compute representatives $C' = \{c'_1, \dots, c'_k\}$ for this partition. Note that $\dim(C') = k - 1$ and that $\dim(C')^{\perp} = \mathcal{D}(j) - k + 1 = \mathcal{D}(j+1)$. We call a color class P_i *light* if $n_i \leq \mathcal{M}(j) - \mathcal{M}(j+1)$; otherwise we call P_i *heavy*. We find $\mathcal{D}(j+1)$ light color classes and project these orthogonally onto $\operatorname{span}(C')^{\perp}$. Let $\widetilde{P}_{j_1}, \dots, \widetilde{P}_{j_{\mathcal{D}(j+1)+1}}$ denote the projections. Next, we recursively compute an $\mathcal{M}(j+1)$-colorful choice $\widetilde{Q}$ for the space orthogonal to $\operatorname{span}(C')$ with $(\widetilde{P}_{j_1}, \dots, \widetilde{P}_{j_{\mathcal{D}(j+1)+1}}, d_0, \mathcal{M}, \mathcal{D})$ as input. Let Q be the point set whose projection gives $\widetilde{Q}$. Using Lemma 2.4, we compute a point c'_j s.t. $\operatorname{conv}(Q \cup C' \setminus c'_j)$ contains the origin. We replace the subset C_j of C by Q and prune C again with Lemma 2.1. Since each representative c'_i is contained in the cone $\operatorname{pos}(C_i)$, $Q \cup C \setminus C_j$ still contains the origin in its convex hull and hence the invariant is maintained. Thus, in one iteration of the algorithm, at least one point from each heavy color class is replaced by points from light color classes. This is repeated until no heavy color classes remain in C. See Algorithm 2.1 for pseudocode.

Theorem 2.7. *Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be sets s.t. $|P_i| \leq d+1$ and s.t. $\vec{0} \in \operatorname{conv}(P_i)$, for $i = 1, \dots, d+1$. Furthermore, let $d_0 \in \mathbb{N}$ be the base case threshold and $\mathcal{M}, \mathcal{D} : \mathbb{N}_0 \to \mathbb{N}$ a pair of feasible parameter functions. On input $(P_1, \dots, P_{d+1}, d_0, \mathcal{M}, \mathcal{D})$, Algorithm 2.1 returns an $\mathcal{M}(0)$-colorful choice. The running time is given by the following recurrence relation:*

$$T(j) \leq \begin{cases} O(d_0^{d_0}), & \text{if } \mathcal{D}(j) \leq d_0, \text{ and} \\ O(\mathcal{D}(j))T(j+1) + O(\mathcal{D}(j)^5), & \text{otherwise.} \end{cases}$$

Proof. We prove correctness by showing that the algorithm respects the parameter functions $\mathcal{D}$ and $\mathcal{M}$. In particular, we prove

Algorithm 2.1: $\varepsilon(d+1)$-apx by Rebalancing

```
   input: $P_1, \dots, P_{d'+1} \subset \mathbb{R}^{d'}$ s.t. $\vec{0} \in \mathrm{conv}(P_i)$ for all $i = 1, \dots, d'+1$, base case threshold
          $d_0 \in \mathbb{N}$, approximation parameter function $\mathcal{M} : \mathbb{N}_0 \to \mathbb{N}$, dimension parameter
          function $\mathcal{D} : \mathbb{N}_0 \to \mathbb{N}$, recursion depth $j$ (initially 0),
 1 if $d' \le d_0$ then
 2     return brute force computed perfect colorful choice
 3 $C \leftarrow P_1$
 4 Prune $C$ with Lemma 2.1.
 5 $d'' \leftarrow \mathcal{D}(j+1)$; $k \leftarrow d' - d'' + 1$
 6 while $C$ is not an $\mathcal{M}(j)$-colorful choice do
 7     Partition $C$ into $k$ sets $C_1, \dots, C_k$ s.t. for all color classes $P_i$ and all pairs of indices
       $1 \le l_1, l_2 \le k$, we have $|\#(P_i \cap C_{l_1}) - \#(P_i \cap C_{l_2})| \le 1$.
 8     Apply Lemma 2.6 to $C_1, \dots, C_k$. Let $C' = \{c'_1, \dots, c'_k\}$ be the set of the representatives.
 9     Find $d''+1$ color classes $P_{j_1}, \dots, P_{j_{d''+1}}$ s.t. $|C \cap P_{j_i}| \le \mathcal{M}(j) - \mathcal{M}(j+1)$.
10     for $i = 1$ to $d''+1$ do
11         $\widetilde{P}_{j_i} \leftarrow$ orthogonal projection of $P_{j_i}$ onto $\mathrm{span}(C')^{\perp}$
12     $Q \leftarrow$ recurse($\widetilde{P}_{j_1}, \widetilde{P}_{j_2}, \dots, \widetilde{P}_{j_{d''+1}}$, $d_0$, $\mathcal{M}$, $\mathcal{D}$, $j+1$)
13     Apply Lemma 2.4 to $C'$ and $Q$ to find a point $c'_i \in C'$ s.t. $\vec{0} \in \mathrm{conv}(Q \cup C' \setminus \{c'_i\})$.
14     $C \leftarrow \left(\bigcup_{j=1, j \ne i}^{k+1} C_j\right) \cup Q$
15     Prune $C$ with Lemma 2.1.
16 return $C$
```

(i) the dimension in the jth recursion is $\mathcal{D}(j)$; and

(ii) in the jth recursion, the resulting colorful choice is an $\mathcal{M}(j)$-colorful choice.

(i) By our assumption, $\mathcal{D}(0) = d$ holds initially. Assume now that we are in recursion level j and that the input points are $\mathcal{D}(j)$ dimensional. In line 11, the point sets for the subproblem are projected onto $\mathrm{span}(C')^{\perp}$. By Lemma 2.6, we have $\dim(C') = k-1$ and thus $\dim(\mathrm{span}(C')^{\perp}) = \mathcal{D}(j) - k + 1 = \mathcal{D}(j+1)$, as desired. Since $\mathcal{D}$ is strictly decreasing, the recursion depth is finite.

(ii) We prove the claim by induction on the recursion depth j. By (i), the base case (i.e., $\mathcal{D}(j) \le d_0$) is reached eventually. The claim trivially holds in the base case, since a perfect colorful choice is always a valid approximation, regardless of $\mathcal{M}$. Assume now that the current recursion depth is j and that the claim holds for all $j' > j$. Let $C^{(t)}$ denote the set C after t iterations of the while-loop in the jth recursion. We show the following invariant:

(α) $\vec{0} \in \mathrm{conv}(C^{(t)})$,

(β) for all color classes P_i, $i = 2, \dots, d+1$, we have $|C^{(t)} \cap P_i| \le \mathcal{M}(j)$, and

(γ) $|C^{(t-1)} \cap P_1| > |C^{(t)} \cap P_1|$, for $t \ge 1$.

The invariant implies that the while-loop terminates and an $\mathcal{M}(j)$-colorful choice is returned.

Before the first iteration, the invariant holds since $C^{(0)} = P_1$. Assume we are now in iteration t and the invariant holds for all previous iterations. Due to Lemmas 2.6 and 2.4, we have $\vec{0} \in \mathrm{conv}(C^{(t)})$ and thus Property (α) holds. By the induction hypothesis, the recursively computed set Q in line 12 is a $\mathcal{M}(j+1)$-colorful choice. Since we use only light color classes in the recursion, adding the points from Q to $C^{(t)}$ does not violate Property (β) of the invariant. It remains to show that we can always find $\mathcal{D}(j+1)+1$ light color classes. Since C is pruned to at most $\mathcal{D}(j)+1$ points at the end of each while-loop iteration, the number of heavy color classes is upper bounded by $\left\lfloor \frac{\mathcal{D}(j)+1}{\mathcal{M}(j)-\mathcal{M}(j+1)} \right\rfloor$. This is at most $\mathcal{D}(j)-\mathcal{D}(j+1)$ by our assumptions. Therefore, there are always at least $\mathcal{D}(j+1)+1$ light color classes.

Finally, we need to check that the number of points from P_1 in $C^{(t)}$ is strictly less than in $C^{(t-1)}$. By our assumptions, $\mathcal{M}(j)+1 \geq \mathcal{D}(j)-\mathcal{D}(j+1)+1 = k$. Since $C^{(t-1)}$ was not a $\mathcal{M}(j)$-colorful choice (otherwise the while-loop would have terminated), $C^{(t-1)}$ contains at least $\mathcal{M}(j)+1$ points from P_1 and thus each set C_i in line 7 contains at least one point from P_1. Since one of these sets is removed in line 14 and Q does not contain the color P_1, Property (γ) of the invariant also holds.

We now analyze the running time. During each iteration of the while-loop, the maximum number of points from each color class is reduced by one until the desired approximation guarantee is reached. Thus, the total number of iterations is bounded by $\mathcal{D}(j)+1-\mathcal{M}(j) = O(\mathcal{D}(j))$. Each iteration of the while-loop requires $O(\mathcal{D}(j)^4)$ time. The recursion stops when the dimensionality is at most d_0. In this case, we compute a perfect colorful choice by brute force in $O(d_0^{d_0})$ time. Thus, we get the claimed recurrence relation:

$$T(j) \leq \begin{cases} O(d_0^{d_0}), & \text{if } \mathcal{D}(j) \leq d_0, \text{ and} \\ O(\mathcal{D}(j))T(j+1) + O(\mathcal{D}(j)^5), & \text{otherwise.} \end{cases}$$

■

Proof of Theorem 1.3. We use Algorithm 2.1 with parameter functions $\mathcal{M}(j) = \lceil \varepsilon(1-\varepsilon)^{j/2}(d+1) \rceil$ and $\mathcal{D}(j) = \lceil (1-\varepsilon)^j(d+1) \rceil$. In particular, we reduce the dimension by $\varepsilon(d+1)$ in each step of the recursion. However, in the jth recursion, we do not compute a $\lceil \varepsilon(\mathcal{D}(j)+1) \rceil$-colorful choice, but a $\lceil (1-\varepsilon)^{-j/2}\varepsilon\mathcal{D}(j) \rceil$-colorful choice. This "slack" increases throughout the recursion. Eventually, the dimension is smaller than the desired approximation guarantee. Then, pruning C with Lemma 2.1 in line 4 already gives a valid approximation.

We first check that $\mathcal{M}$ and $\mathcal{D}$ are feasible: 1. and 2. hold trivially. It remains to prove Condition 3. We have

$$\begin{aligned} \left\lfloor \frac{\mathcal{D}(j)+1}{\mathcal{M}(j)-\mathcal{M}(j+1)} \right\rfloor &= \left\lfloor \frac{\lceil (1-\varepsilon)^j(d+1) \rceil + 1}{\lceil \varepsilon(1-\varepsilon)^{j/2}(d+1) \rceil - \lceil \varepsilon(1-\varepsilon)^{(j+1)/2}(d+1) \rceil} \right\rfloor \\ &\leq \frac{(1-\varepsilon)^j(d+1)+2}{\varepsilon(1-\varepsilon)^{j/2}(d+1) - \varepsilon(1-\varepsilon)^{(j+1)/2}(d+1) - 1} \\ &\leq \frac{(1-\varepsilon)^{j/2}(d+1)}{\varepsilon^2(1-\varepsilon)^{j/2}(d+1)\left[1-\sqrt{1-\varepsilon}-\frac{1}{\varepsilon(1-\varepsilon)^j(d+1)}\right]} \\ &\leq \frac{1}{\varepsilon^2\left[\frac{\varepsilon}{2}-\frac{1}{\varepsilon\mathcal{D}(j)-1}\right]} \end{aligned}$$

and

$$\begin{aligned} \mathcal{D}(j)-\mathcal{D}(j+1) &= \lceil (1-\varepsilon)^j(d+1) \rceil - \lceil (1-\varepsilon)^{j+1}(d+1) \rceil \\ &\geq \varepsilon(1-\varepsilon)^j(d+1) - 1 \\ &\geq \mathcal{D}(j) - 2. \end{aligned}$$

Furthermore, for $\mathcal{D}(j) \geq 4(1+\varepsilon)/\varepsilon^3$, we have

$$\frac{1}{\varepsilon^2\left[\frac{\varepsilon}{2} - \frac{1}{\varepsilon\mathcal{D}(j)-1}\right]} \leq \mathcal{D}(j) - 2.$$

So for $\mathcal{D}(j) = \Omega(1/\varepsilon^3)$, we have

$$\left\lfloor \frac{\mathcal{D}(j)+1}{\mathcal{M}(j) - \mathcal{M}(j+1)} \right\rfloor \leq \mathcal{D}(j) - \mathcal{D}(j+1).$$

Thus, Condition 3. holds if $d_0 = \Omega(1/\varepsilon^3)$.

It remains to analyze the running time. The recursion stops as soon as $\mathcal{M}(j) \geq \mathcal{D}(j)+1$. Then, the while-loop is skipped since pruning P_1 with Lemma 2.1 already gives a valid approximation. Since $\mathcal{M}(j) \geq \varepsilon(1-\varepsilon)^{j/2}(d+1)$ and $3(1-\varepsilon)^j(d+1) \geq \mathcal{D}(j)+1$, we have $\mathcal{M}(j) \geq \mathcal{D}(j)+1$ for $j = O((1/\varepsilon)\log(1/\varepsilon))$, using the fact that $-\log(1-\varepsilon) \geq \varepsilon$. Thus, we obtain the following recurrence relation for the running time:

$$T(j) \leq \begin{cases} O(d^4), & \text{if } j = \Omega(\varepsilon^{-1}\log\varepsilon^{-1}) \\ O(\lceil (1-\varepsilon)^j(d+1)\rceil)T(j+1) + O(\mathcal{D}(j)^5), & \text{otherwise.} \end{cases}$$

This solves to the claimed running time $d^{O((1/\varepsilon)\log(1/\varepsilon))}$. ■

2.3 Varying the Number of Color Classes

So far, we assumed that we have $d+1$ color classes as input. Now, we explore the consequences of varying this parameter. First, we show that given $\Omega(d^2 \log d)$ color classes, a generalization of the algorithm from Proposition 2.3 can compute a perfect colorful choice in time $d^{O(\log d)}$, improving the brute force $d^{O(d)}$ algorithm asymptotically. Second, we look at the setting where we have only two color classes. In this case the colorful Carathéodory theorem guarantees the existence of a $\lceil (d+1)/2 \rceil$-colorful choice that contains the origin in its convex hull. We show how to compute a $(d - \Theta(\sqrt{d}))$-colorful choice in time $O(d^4)$.

We begin with the proof of Theorem 1.4. The algorithm follows the structure of Miller and Sheehy's algorithm for computing approximate Tverberg partitions [9]. We repeatedly combine $d+1$ m-colorful choices (for some m) to one $\lceil m/2 \rceil$-colorful choice. Eventually, we obtain a perfect colorful choice.

Lemma 2.8. *Let $C_1, \dots, C_{d+1} \subset \mathbb{R}^d$ be m-colorful choices of size at most $d+1$. Then, a $\lceil m/2 \rceil$-colorful choice can be computed in $O(d^5)$ time.*

Proof. Similar to the proof of Proposition 2.3, we partition each color class C_i $(i = 1, \dots, d+1)$ into two sets $C_{i,1}, C_{i,2}$ of equal size, however this time not arbitrary: for each color P_j that appears in C_i, the points in $C_i \cap P_i$ are distributed evenly among both sets $C_{i,1}$ and $C_{i,2}$. Now, the proof of Proposition 2.3 states that we can find in $O(d^5)$ time a set C that contains the origin in its convex hull and that contains exactly one of the two sets $C_{i,1}, C_{i,2}$, for $i = 1, \dots, d+1$. Since all sets C_i are m-colorful choices, C is a $\lceil m/2 \rceil$-colorful choice. ■

Proof of Theorem 1.4. Let A be an array of size $k = \Theta(\log d)$. We store in A approximate colorful choices, where all approximate colorful choices in one cell of A have the same approximation guarantee. We maintain the following invariant: if all sets in $A[i-1]$ are m-colorful choices, then all sets in $A[i]$ are $\lceil m/2 \rceil$-colorful choices. Furthermore, each color appears only in a single set.

Initially, $A[0]$ contains all $\Theta(d^2 \log d)$ color classes. The invariant holds since all other cells of A are empty. Let c_i denote the approximation guarantee of the colorful choices in $A[i]$. We have $c_0 = d+1$ since each color class is trivially a $(d+1)$-colorful choice and thus $c_1 = \lceil (d+1)/2 \rceil$, $c_2 = \lceil c_1/2 \rceil = \lceil \lceil (d+1)/2 \rceil /2 \rceil$, and so on. While we have not computed a perfect colorful choice, we repeat the following steps: let i be the maximum index s.t. $A[i]$ contains at least $d+1$ sets and let $C_1, \dots, C_{d+1}$ be $d+1$ arbitrary sets from $A[i]$. We apply Lemma 2.8 to obtain one c_{i+1}-colorful choice C. Let C' be the set that we obtain by pruning C with Lemma 2.1. If C' is a perfect colorful choice, we return it. Otherwise, we add it to $A[i+1]$. Furthermore, we add all colors that were removed during the pruning, i.e., colors that appear in C but not in C', to $A[0]$ as these colors do not appear in any set stored in A. Thus, the invariant is maintained.

We claim that a combination of $d+1$ sets in $A[k]$ for $k = \lceil \log(d+1) \rceil + 1$ results in a perfect colorful choice. We have

$$c_j = \lceil \dots \lceil \lceil (d+1) \underbrace{/2 \rceil /2 \rceil \dots \rceil}_{k \text{ times}} \leq \frac{d+1}{2^k} + \sum_{i=0}^{k-1} 2^{-i} \leq \frac{d+1}{2^k} + 2.$$

Thus, sets in $A[\lceil \log(d+1) \rceil]$ are 3-colorful choices, sets in $A[\lceil \log(d+1) \rceil + 1] = A[k]$ are 2-colorful choices, and the combination of $d+1$ sets in $A[k]$ gives a perfect colorful choice.

It remains to show that we can always find an index i s.t. $A[i]$ contains at least $d+1$ sets. Suppose there is no such index. Then, each cell $A[i]$ contains at most d sets and each set contains at most d colors since it is not a perfect colorful choice. Hence, at most $d^2 k = d^2(\lceil \log(d+1) \rceil + 1)$ colors can appear in A. Thus, if we have $d^2 k + 1 = \Theta(d^2 \log d)$ colors, progress is always possible.

Let us consider the running time. Let $T(i)$ denote the time to compute a set in level i of A. One combination step takes $O(d^5)$ time, both for the applications of Lemmas 2.8 and 2.1. To compute a set in level i, we have to compute $d+1$ sets in level $i-1$. This results in the following recurrence relation:

$$T(i) = \begin{cases} O(1), & \text{if } i = 0 \\ (d+1)T(i-1) + O(d^5), & \text{otherwise.} \end{cases}$$

Thus, computing one set in level $k+1$ takes $d^{O(\log d)}$ time. ■

Proof of Proposition 1.5. Let P and Q be the two color classes. Let k be a parameter to be determined later. We prune P with Lemma 2.1 and partition it into k sets $P_1, \dots, P_k$ of equal size. We apply Lemma 2.6 to obtain representatives $P' = \{p'_1, \dots, p'_k\}$ for these sets and project Q onto the $(d-k+1)$-dimensional subspace $\operatorname{span}(P')^{\perp}$. Again, we prune Q with Lemma 2.1 and apply Lemma 2.4 to replace one point p'_i of P' with Q. Thus, the set $C = \bigcup_{j=1, j \neq i}^{k} P_i \cup Q$ contains the origin its convex hull and has at most $\max\{\lceil (d+1)(1-1/k) \rceil, d-k+1\}$ points of each color. Setting $k = \Theta(\sqrt{d})$ gives the result. ■

3 The Nearest Colorful Polytope Problem

The complexity class *Polynomial-Time Local Search* (PLS) contains local search problems for which a single improvement step can be carried out in polynomial-time. In contrast to complexity classes for decision problems such as P and NP, the existence of a solution (a local optimum) to a PLS problem is always guaranteed. Instead, the difficulty lies in finding the solution. Mathematically, a PLS problem A is a relation $A \subseteq \mathcal{I} \times \mathcal{S}$, where $\mathcal{I}$ is the set of *problem instances* and $\mathcal{S}$ is the set of *candidate solutions*. The relation A is in PLS if:

- problem instances $I \in \mathcal{I}$ and candidate solutions $s \in \mathcal{S}$ are polynomial-time verifiable and the size of the valid candidate solutions for an instance I is polynomial in the size of I;
- there is a polynomial-time computable function $\mathcal{B} : \mathcal{I} \to \mathcal{S}$ that returns some candidate solution (the base solution) for each instance;
- there is a polynomial-time computable function $\mathcal{C} : \mathcal{I} \times \mathcal{S} \to \mathbb{N}$ that assigns *costs* to each instance-solution pair;
- there is a polynomial-time computable neighborhood function $\mathcal{N} : \mathcal{I} \times \mathcal{S} \to 2^{\mathcal{S}}$ assigning each candidate solution a set of neighboring candidate solutions; and
- for every instance $I \in \mathcal{I}$, A contains exactly the pairs (I, s) so that s is a *local optimum* for I; i.e., all elements in $\mathcal{N}(I, s)$ have smaller costs in a maximization problem and larger costs in a minimization problem.

The computational problem modeled by A is: given $I \in \mathcal{I}$, find an $s \in \mathcal{S}$ s.t. $(I, s) \in A$. The following algorithm is called the *standard algorithm*: start with the base solution $\mathcal{B}(I)$ and use $\mathcal{N}$ to improve until a local optimum is reached. Each iteration takes polynomial time, but the total number of iterations may be exponential. There are examples where it is PSPACE-hard to find the solution given by the standard algorithm [1, Chapter 2].

To define hardness with respect to PLS, we need an appropriate reduction concept. A *PLS-reduction* from a PLS-problem A to a PLS-problem B is given by two polynomial-time computable functions $f : \mathcal{I}_A \to \mathcal{I}_B$ and $g : \mathcal{I}_A \times \mathcal{S}_B \to \mathcal{S}_A$ such that f maps A-instances to B-instances and g maps local optima for B to local optima for A. Thus, if A is PLS-reducible to B, we can convert any algorithm for B into an algorithm for A with polynomial-time overhead. We call B *PLS-complete* if all problems in PLS are PLS-reducible to B.

Like PPAD, PLS is a subset of the class *Total Function NP* (TFNP). TFNP contains search problems whose solution can be verified in polynomial time. No problem in TFNP can be NP-hard unless NP = coNP [4]. On the other hand, it is not believed that PLS-complete problems can be solved in polynomial-time, although this would not break any assumptions on complexity classes. For more information see one of the several main publications on the topic [1, 8, 12, 4]. In the language of PLS, L-NCP is defined as follows:

Definition 3.1 (L-NCP)**.**

Instances $\mathcal{I}_{\mathsf{NCP}}$. Set families $P = \{P_1, \ldots, P_n\}$ in $\mathbb{R}^d$, where each $P_i \subset \mathbb{R}^d$ is a color.

Solutions $\mathcal{S}_{\mathsf{NCP}}$. All perfect colorful choices, i.e., sets with exactly one point of each color.

Cost function $\mathcal{C}_{\mathsf{NCP}}$. Let S_{NCP} be a colorful choice. Then, $\mathcal{C}(S_{\mathrm{NCP}}) = \| \operatorname{conv}(S_{\mathrm{NCP}})\|_1$, where $\| \operatorname{conv}(S_{\mathrm{NCP}})\|_1 = \min\{\|q\|_1 \mid q \in \operatorname{conv}(S_{\mathrm{NCP}})\}$. We want to minimize $\mathcal{C}_{\mathrm{NCP}}$.

Neighborhood $\mathcal{N}_{\mathsf{NCP}}$. The neighbors $\mathcal{N}(S_{\mathrm{NCP}})$ of a colorful choice S_{NCP} are all colorful choices that can be obtained by swapping one point with another point of the same color.

We reduce the following PLS-complete problem [12, Corollary 5.12] to L-NCP:

Definition 3.2 (Max-2SAT/Flip)**.**

Instances $\mathcal{I}_{\mathsf{M2SAT}}$. All weighted 2-CNF formulas $\bigwedge_{i=1}^{d} C_i$, where each clause C_i is the disjunction of at most two literals and has weight $w_i \in \mathbb{N}_+$.

Solutions $\mathcal{S}_{\mathsf{M2SAT}}$. Let $x_1, x_2, \ldots, x_n$ be the variables appearing in the clauses. Then, every complete assignment $\mathcal{A} : \{x_1, \ldots, x_n\} \to \{0, 1\}$ of these variables is a solution.

Cost function $\mathcal{C}_{\mathsf{M2SAT}}$. The cost of an assignment is the sum of the weights of all satisfied clauses. We want to maximize the cost function.

Neighborhood $\mathcal{N}_{\mathsf{M2SAT}}$. The neighbors $\mathcal{N}(\mathcal{A})$ of an assignment $\mathcal{A}$ are all assignments obtained by flipping (i.e., negating) a single variable in $\mathcal{A}$.

Proof of Theorem 1.6. Let $I_{\mathrm{M2SAT}} = (C_1, \dots, C_d, w_1, \dots, w_d, x_1, \dots, x_n)$ be an instance of M2SAT. We construct an instance I_{NCP} of L-NCP in which each colorful choice encodes an assignment to the variables in I_{M2SAT}. Furthermore, the distance to the origin of the convex hull of a colorful choice in I_{NCP} will be the total weight of all unsatisfied clauses of the encoded assignment for I_{M2SAT}.

For each variable x_i, we introduce a color class $P_i = \{p_i, \overline{p_i}\}$ consisting of two points in $\mathbb{R}^d$ that encode whether x_i is set to 1 or 0. We assign the jth dimension to the jth clause and set $(p_i)_j = -nw_j$, if $x_i = 1$ satisfies clause j, and $(p_i)_j = w_j$, otherwise. Similarly, $(\overline{p_i})_j = -nw_j$, if $x_i = 0$ satisfies C_j, and $(\overline{p_i})_j = w_j$ otherwise. A colorful choice S of $P_1, \dots, P_n$ corresponds to the assignment in I_{M2SAT} where x_i is 1 if $p_i \in S$ and 0 if $\overline{p_i} \in S$. More formally, we define a mapping $g : \mathcal{I}_{\mathrm{M2SAT}} \times \mathcal{S}_{\mathrm{NCP}} \to \mathcal{S}_{M2SAT}$ between the solutions of the L-NCP instance and the M2SAT instance in the following way:

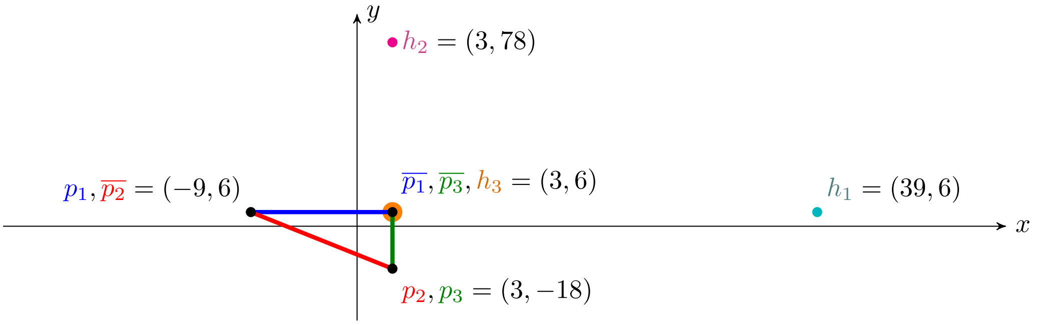
$$g(I_{\mathrm{M2SAT}}, S_{\mathrm{NCP}})(x_i) = \begin{cases} 1, & \text{if } p_i \in S_{\mathrm{NCP}} \\ 0, & \text{if } \overline{p_i} \in S_{\mathrm{NCP}}. \end{cases}$$

The main idea is to construct an instance of L-NCP in which the convex hull of a colorful choice S contains the origin if projected onto the dimensions corresponding to the satisfied clauses. Furthermore, if projected onto the subspace corresponding to the unsatisfied clauses, the distance of $\mathrm{conv}(S)$ to the origin will be equal to the total weight of those clauses.

We introduce additional helper color classes to decrease the distance to the origin in dimensions that correspond to satisfied clauses. In particular, we have for each clause C_j a color class $H_j = \{h_j\}$ consisting of a single point, where

$$(h_j)_k = \begin{cases} (d+1)\left((n+2) - \frac{d}{d+1}\right) w_j, & \text{if } k = j \\ w_k, & \text{otherwise.} \end{cases}$$

The last helper color class $H_{d+1} = \{h_{d+1}\}$ again contains a single point, but now all coordinates are set to the clause weights, i.e., $(h_{d+1})_j = w_j$ for $j = 1, \dots, d$. See Fig. 2.



y
$h_2 = (3, 78)$
$p_1, \overline{p_2} = (-9, 6)$
$\overline{p_1}, \overline{p_3}, h_3 = (3, 6)$
$h_1 = (39, 6)$
x
$p_2, p_3 = (3, -18)$

Fig. 2: Construction of the point sets corresponding to the M2SAT instance $(x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_3)$ with weights 3 and 6, respectively.

The remaining proof is divided into two parts: (i) for every colorful choice S_{NCP} of the L-NCP problem instance $\{P_1, \dots, P_n, H_1, \dots, H_{d+1}\}$, the cost $\mathcal{C}_{\text{NCP}}(S_{\text{NCP}})$ is lower-bounded by the total weight of unsatisfied clauses in $g(S_{\text{NCP}})$; and (ii) this lower bound is tight, i.e., the distance of the convex hull of any colorful choice S_{NCP} to the origin is at most the total weight of unsatisfied clauses in $g(S_{\text{NCP}})$.

Both claims together imply that $\mathcal{C}_{\text{NCP}}(S_{\text{NCP}})$ equals the total weight of unsatisfied clauses for the assignment $g(S_{\text{NCP}})$, which proves the theorem: consider some local optimum S^*_{NCP} of the L-NCP instance. By definition, the costs of all other colorful choices that can be obtained from S^*_{NCP} by exchanging one point with another of the same color are greater or equal to $\mathcal{C}_{\text{NCP}}(S^*_{\text{NCP}})$. That is, the total weight of unsatisfied clauses in $g(S^*_{\text{NCP}})$ cannot be decreased by flipping a variable, which is equivalent to $g(S^*_{\text{NCP}})$ being a local optimum of the M2SAT instance.

(i) Let S_{NCP} be a colorful choice and assume some clause C_j is not satisfied by $g(S_{\text{NCP}})$. By construction, the jth coordinate of each point q in S_{NCP} is at least w_j. Thus, the jth coordinate of every convex combination of the points in S_{NCP} is at least w_j. This implies (i).

(ii) Given a colorful choice S_{NCP}, we construct a convex combination of S_{NCP} that gives a point p whose distance to the origin is exactly the total weight of unsatisfied clauses in $g(S_{\text{NCP}})$. Let in the following part A_k denote the set of clauses C_j that are satisfied by k literals w.r.t. $g(S_{\text{NCP}})$, for $k = 0, 1, 2$. As a first step towards constructing p, we show the existence of an intermediate point in the convex hull of the helper classes:

Lemma 3.3. *There is a point $h \in \operatorname{conv}(H_1, \dots, H_{d+1})$ whose jth coordinate is $(n+2)w_j$ if $j \in A_2$ and w_j otherwise.*

Proof. Take $h = \sum_{a \in A_2} \frac{1}{d+1} h_a + \left(1 - \frac{|A_2|}{d+1}\right) h_{d+1}$. Then, for $j \in A_0 \cup A_1$, we have

$$(h)_j = \sum_{a \in A_2} \frac{1}{d+1}(h_a)_j + \left(1 - \frac{|A_2|}{d+1}\right)(h_{d+1})_j \overset{j \notin A_2}{=} \sum_{a \in A_2} \frac{1}{d+1} w_j + \left(1 - \frac{|A_2|}{d+1}\right) w_j = w_j.$$

And for $j \in A_2$, we have

$$\begin{aligned} (h)_j &= \sum_{a \in A_2} \frac{1}{d+1}(h_a)_j + \left(1 - \frac{|A_2|}{d+1}\right)(h_{d+1})_j \\ &= \frac{1}{d+1} h_j + \sum_{a \in A_2 \setminus \{j\}} \frac{1}{d+1}(h_a)_j + \left(1 - \frac{|A_2|}{d+1}\right)(h_{d+1})_j \\ &= \left((n+2) - \frac{d}{d+1}\right) w_j + \frac{d}{d+1} w_j = (n+2)w_j, \end{aligned}$$

as desired. ■

Let $l_i \in P_i$ be the point from P_i in S_{NCP}. Consider $p = \sum_{i=1}^n \frac{1}{n+1} l_i + \frac{1}{n+1} h$. We show that $(p)_j = w_j$, for $j \in A_0$, and $(p)_j = 0$, otherwise. Let us start with $j \in A_0$. Since $g(S_{\text{NCP}})$ does not satisfy C_j, the jth coordinate of the points $l_1, \dots, l_n$ is w_j. Also, $(h)_j = w_j$, by Lemma 3.3. Thus, $(p)_j = w_j$. Consider now some $j \in A_1$ and let b be s.t. the point l_b corresponds to the single literal that satisfies C_j.

$$\begin{aligned} (p)_j &= \sum_{i=1}^n \frac{1}{n+1}(l_i)_j + \frac{1}{n+1}(h)_j \\ &= \frac{1}{n+1}(l_b)_j + \sum_{i=1, i \neq b}^n \frac{1}{n+1}(l_i)_j + \frac{1}{n+1}(h)_j = \frac{-n}{n+1} w_j + \frac{n}{n+1} w_j = 0. \end{aligned}$$

Finally, consider some $j \in A_2$ and let b_1, b_2 be the indices of the two literals that satisfy C_j:

$$\begin{aligned}(p)_j &= \sum_{i=1}^{n} \frac{1}{n+1}(l_i)_j + \frac{1}{n+1}(h)_j \\ &= \frac{1}{n+1}(l_{b_1})_j + \frac{1}{n+1}(l_{b_2})_j + \sum_{i=1, i \notin \{b_1,b_2\}}^{n} \frac{1}{n+1}(l_i)_j + \frac{1}{n+1}(h)_j \\ &= \frac{-2n}{n+1}w_j + \frac{n-2}{n+1}w_j + \frac{n+2}{n+1}w_j = 0\end{aligned}$$

This concludes the proof of (ii). ∎

Proof of Theorem 1.7. The proof of Theorem 1.6 can be adapted easily to reduce 3SAT to G-NCP: given a set of clauses $C_1, \ldots, C_d$, we set the weight of each clause to 1 and construct the same point sets as in the PLS reduction. Additionally, we introduce for each clause C_j a new helper color class $H'_j = \{h'_j\}$, where

$$(h'_i)_j = \begin{cases} (d+1)\left((2n+2) - \frac{d}{d+1}\right), & \text{if } i = j, \text{ and} \\ 1, & \text{otherwise.} \end{cases}$$

Let S now be any colorful choice and $A = g(S)$ the corresponding assignment. As in the PLS-reduction, we define the sets A_k, $k = 0, \ldots, 3$, to contain all clauses that are satisfied by exactly k literals in the assignment A. Then, the following point h is contained in the convex hull of the helper points:

$$h = \sum_{a \in A_2} \frac{h_a}{d+1} + \sum_{a' \in A_3} \frac{h'_{a'}}{d+1} + \left(1 - \frac{|A_2|}{d+1}\right) h_{d+1}.$$

Again, the convex combination $p = \sum_{i=1}^{n} \frac{1}{n+1} l_i + \frac{1}{n+1} h$ results in a point in the convex hull of S whose distance to the origin is the number of unsatisfied clauses, where $l_i \in P_i$ denotes the point from P_i that is contained in S. Together with Claim **(i)** from the proof of Theorem 1.6, 3SAT can be decided by knowing a global optimum S^* to the NCP problem: if the distance from $\operatorname{conv}(S^*)$ to the origin is 0, $g(S^*)$ is a satisfying assignment. If not, there exists no satisfying assignment at all. ∎

4 Conclusion

We have proposed a new notion of approximation for the colorful Carathéodory theorem and presented an abstract approximation scheme. By choosing the parameters carefully, we could obtain a polynomial-time algorithm that computes $\lceil \varepsilon(d+1) \rceil$-colorful choices for any constant $\varepsilon > 0$. One of the key motivations for studying this kind of approximation was the tight connection to approximating Tverberg's theorem. Unfortunately, if we convert the algorithm from Theorem 1.3 to an approximation algorithm for Tverberg using Sarkaria's proof, we obtain an algorithm with a trivial approximation guarantee. The approximation guarantee of the algorithm from Theorem 1.3 is right at the threshold to get a nontrivial Tverberg approximation algorithm: any efficient algorithm computing an $d^{o(1)}$-colorful choice would result in an efficient approximation algorithm for Tverberg's theorem with a nontrivial approximation guarantee. This is particularly interesting as no nontrivial efficient approximating algorithm for Tverberg's theorem is known. The existence of such an algorithm was conjectured by Miller and Sheehy [9]. However, it does not seem possible to get an approximation algorithm for Tverberg that is polynomial in d and *linear* in the number of points with this approach, as in this case even pruning the color classes with Carathéodory's theorem would require too much time.

In the second part, we have studied the complexity of a natural generalization of the colorful Carathéodory theorem, the Nearest Colorful Polytope problem, in two settings: first, we have proved that the corresponding local search problem is PLS-complete by a reduction to Max2SAT. Using an adaptation of the PLS-reduction, we could prove that the problem becomes NP-hard if we restrict the solutions to global optima. Although the PLS-completeness of the Nearest Colorful Polytope problem together with Bárány's proof indicate that PLS is the right complexity class to show hardness of the Colorful Carathéodory problem, there is a striking difference between the Colorful Carathéodory problem and any known PLS-complete problem: the costs of local optima are known a-priori. While a PLS-complete problem with this property would not lead to a contradiction, this creates a major stumbling block in the construction of a reduction.

Acknowledgements We would like to thank Fréderic Meunier and Pauline Sarrabezolles for interesting discussions on the Colorful Carathéodory problem and for hosting us during a research stay at the École Nationale des Ponts et Chaussées.

References

[1] Emile Aarts and Jan Karel Lenstra, editors. *Local search in combinatorial optimization.* Princeton University Press, 2003.

[2] Imre Bárány. A generalization of Carathéodory's theorem. *Discrete Math.*, 40(2–3):141–152, 1982.

[3] Imre Bárány and Shmuel Onn. Colourful linear programming and its relatives. *Math. Oper. Res.*, 22(3):550–567, 1997.

[4] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *J. Comput. System Sci.*, 37(1):79–100, 1988.

[5] Jiří Matoušek. *Lectures on discrete geometry.* Springer, 2002.

[6] Frédéric Meunier and Antoine Deza. A further generalization of the colourful Carathéodory theorem. In *Discrete geometry and optimization*, volume 69 of *Fields Inst. Commun.*, pages 179–190. Springer, New York, 2013.

[7] Frédéric Meunier and Pauline Sarrabezolles. Colorful linear programming, Nash equilibrium, and pivots. *arxiv:1409.3436*, 2014.

[8] Wil Michiels, Emile Aarts, and Jan Korst. *Theoretical aspects of local search.* Monographs in Theoretical Computer Science. An EATCS Series. Springer, Berlin, 2007.

[9] Gary L. Miller and Donald R. Sheehy. Approximate center points with proofs. In *Computational geometry (SCG'12)*, pages 303–309. ACM, New York, 2012.

[10] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. System Sci.*, 48(3):498–532, 1994.

[11] Karanbir S. Sarkaria. Tverberg's theorem via number fields. *Israel J. Math.*, 79(2–3):317–320, 1992.

[12] Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM J. Comput.*, 20(1):56–87, 1991.

[13] Helge Tverberg. Further generalization of Radon's theorem. *J. London Math. Soc.*, 43:352–354, 1968.