The Pareto Cover Problem

Bento Natura

London School of Economics and Political Science, UK

Meike Neuwohner

Forschungsinstitut für Diskrete Mathematik, Universität Bonn, Germany

Stefan Weltge

Technische Universität München, Germany

Abstract

We introduce the problem of finding a set B of k points in $[0,1]^n$ such that the expected cost of the cheapest point in B that dominates a random point from $[0,1]^n$ is minimized. We study the case where the coordinates of the random points are independently distributed and the cost function is linear. This problem arises naturally in various application areas where customers' requests are satisfied based on predefined products, each corresponding to a subset of features. We show that the problem is NP-hard already for $k = 2$ when each coordinate is drawn from $\{0,1\}$, and obtain an FPTAS for general fixed k under mild assumptions on the distributions.

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures; Theory of computation → Dynamic programming; Theory of computation → Packing and covering problems; Theory of computation → Problems, reductions and completeness; Mathematics of computing → Discrete optimization

Keywords and phrases Pareto, Covering, Optimization, Approximation Algorithm

Digital Object Identifier 10.4230/LIPIcs.ESA.2022.80

Related Version *Full Version*: `https://arxiv.org/abs/2202.08035`

1 Introduction

Let $f\colon [0,1]^n \to \mathbb{R}$ be a continuous cost function and $B \subseteq [0,1]^n$ be a finite set. We consider the function

$$f_B\colon [0,1]^n \to \mathbb{R} \cup \{\infty\}, \quad f_B(x) := \min\{f(b) : x \leqslant b,\, b \in B\},$$

where we write $x \leqslant b$ if x is less or equal than b in every coordinate and say that b *covers* x. In other words, $f_B(x)$ is the smallest cost needed to cover x with a point from B. We say that B is a *Pareto cover*[1] of a probability measure μ on $[0,1]^n$ if a random point can be covered by at least one point from B almost surely, i.e., $\mu(\{x \in [0,1]^n : x \leqslant b \text{ for some } b \in B\}) = 1$. Note that B is always a Pareto cover if it contains the all-ones vector $\mathbf{1}$ (but for some probability measures, B is not required to contain $\mathbf{1}$).

Given f, μ, and an integer $k \geqslant 1$, we study the problem of finding a Pareto cover B of μ with $|B| = k$ such that $\mathrm{E}_\mu[f_B]$ is minimized. That is, we are searching for a Pareto cover B of predefined size such that the expected cost of covering a random point with a point from B is smallest possible.

As an illustration, imagine a city with tourist attractions $[n] := \{1,\dots,n\}$ and suppose that the city wants to design k books B of vouchers for subsets of these attractions. Each tourist x will pick the cheapest book $b \in B$ that covers all attractions that x wants to visit. We can think of x as a binary vector in $\{0,1\}^n$. Assuming that we have some probability

[1] In the context of multi-objective optimization, $x \leqslant b$ is commonly referred to as b *Pareto-dominates* x.

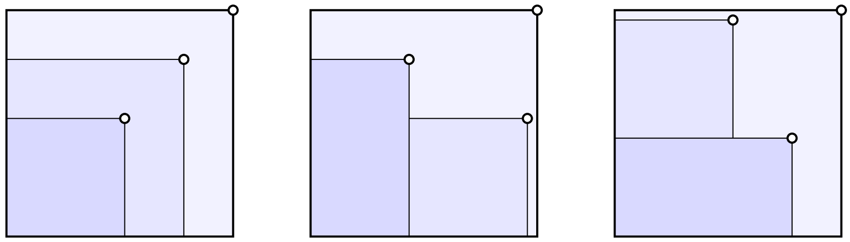

Figure 1 For the Lebesgue measure (uniform distribution) on $[0,1]^2$ with cost $f(x_1,x_2) = x_1 + x_2$, the optimal Pareto covers of size $k = 3$ are $\{(^{12}/_{23}, ^{12}/_{23}), (^{18}/_{23}, ^{18}/_{23}), (1,1)\}$, $\{(^{10}/_{23}, ^{18}/_{23}), (^{22}/_{23}, ^{12}/_{23}), (1,1)\}$, and $\{(^{18}/_{23}, ^{10}/_{23}), (^{12}/_{23}, ^{22}/_{23}), (1,1)\}$.

distribution over the tourists x, we want to determine k books that, in expectation, cover the tourists requests in the cheapest way. Note that, in this example, μ is a discrete measure on $\{0,1\}^n$. Defining an appropriate cost function, an optimal Pareto cover of size k is attained by a set of vectors in $\{0,1\}^n$, each corresponding to a book.

Similar applications can be given for other areas where customer's requests are satisfied based on predefined products, each corresponding to a subset of features. Note that our model also allows for non-binary requests $x \in [0,1]^n \setminus \{0,1\}^n$, which may correspond to features that are available in different quality ranges.

For another application, imagine a gang of robbers that wants to steal paintings in an art gallery. To this end, every gang member studies one painting $i \in [n]$ and estimates the probability p_i of being able to steal it. Their boss decides in advance which subset $S \subseteq [n]$ of paintings to steal. If all corresponding gang members are successful (assuming that they act independently), then the gang will receive a value $v(S)$. Otherwise, they all get caught and the gang receives $v(\emptyset) = 0$. The problem of finding a subset of paintings that maximizes the expected return can be modeled within the above framework as follows. Let μ be the probability measure on $\{0,1\}^n$ that corresponds to setting each coordinate independently to 1 with probability $1 - p_i$, and set $f(x) := v([n]) - v(\{i \in [n] : x_i = 0\})$ for all $x \in \{0,1\}^n$. Denoting by $B^* = \{b^*, \mathbf{1}\}$ an optimal Pareto cover of size $k = 2$, it is easy to see that $S = \{i \in [n] : b^*_i = 0\}$ maximizes the expected return.

Determining optimal Pareto covers of a given size is a difficult problem. Finding an analytical solution seems to be non-trivial even for very basic probability measures and cost functions, see Figure 1. In this work, we study the problem from the point of view of complexity theory. We particularly focus on product measures and linear cost functions, for which the problem is already hard as our first results show.

Here, for $A = \{a_0, \ldots, a_{M+1}\}$ with $0 = a_0 < a_1 < \cdots < a_M < a_{M+1} = 1$ and a vector $p = \big((p^i_\ell)_{\ell=0}^{M+1}\big)_{i=1}^n \in [0,1]^{(M+2)\times n}$ with $\sum_{\ell=0}^{M+1} p^i_\ell = 1$ for all $i \in [n]$, let $\mu_{A,p}$ denote the *discrete product measure* on $[0,1]^n$ where x_i assumes a_ℓ with probability p^i_ℓ, that is,

$$\mu_{A,p}(\{(a_{\ell_i})_{i=1}^n\}) = \prod_{i=1}^n p^i_{\ell_i} \text{ for all } (a_{\ell_i})_{i=1}^n \in A^n. \tag{1}$$

In addition, for $A = \{0,1\}$ and $p \in [0,1]^n$, we define the *binary product measure* $\mu_p := \mu_{\{0,1\},q}$ where $q^i_0 = 1 - p_i$ and $q^i_1 = p_i$ for $i \in [n]$, that is, x_i is set to $a_1 = 1$ with probability p_i, and to $a_0 = 0$ with probability $1 - p_i$. Finally, for a linear cost function f given by $f(x) = c^\intercal x$ for some $c \in \mathbb{R}^n$ and a finite set B, we overload our notation $c_B := f_B$.

The Pareto cover problem turns out to be computationally difficult even in a setting where we restrict ourselves to binary product measures and linear cost functions.

▶ **Theorem 1.** *Let $k \in \mathbb{Z}$, $k \geqslant 2$. Given $p \in ([0,1] \cap \mathbb{Q})^n$, $c \in \mathbb{Q}^n$, $\gamma \in \mathbb{Q}$, the problem of deciding whether there is some Pareto cover B of μ_p such that $|B| = k$ and $\mathrm{E}_{\mu_p}[c_B] \leqslant \gamma$ is weakly NP-complete for k constant. Moreover, there are values of $k \in \Theta(n)$ for which it is strongly NP-hard.*

If the size of the Pareto cover is part of the input, we do not know whether the corresponding problem is in NP. In fact, computing the objective value of a single Pareto cover is already difficult:

▶ **Proposition 2.** *Given a Pareto cover B for the uniform distribution μ on $\{0,1\}^n$, the problem of computing $\mathrm{E}_\mu[\mathbf{1}_B]$ is #P-hard.*

Even when k is constant, it is not immediate how to determine, in polynomial time, the objective value $\mathrm{E}_{\mu_p}[c_B]$ a certain Pareto cover $B = \{b^1, \ldots, b^k\}$ attains for given probabilities $p \in ([0,1] \cap \mathbb{Q})^n$ and a positive cost vector $c \in \mathbb{Q}^n$. In this respect, observe that it is infeasible to simply sum over all vectors $x \in \{0,1\}^n$ since there is an exponential number of them. However, for constant k, we can afford to iterate over all subsets of $[k]$ instead. Moreover, it is not hard to see that for every $J \subseteq [k]$, c_B is constant on the set X_J consisting of all vectors $x \in [0,1]^n$ that are covered precisely by those vectors b^j for which $j \in J$.

We show how to employ dynamic programming in order to compute the values $\mu(X_J)_{J \subseteq [k]}$ for arbitrary discrete product measures μ on $[0,1]^n$. Using this observation as a starting point, we manage to derive a fully polynomial-time approximation scheme (FPTAS) for the case where k is constant.

▶ **Theorem 3.** *Let $k \in \mathbb{N}$ be fixed. Given a discrete product measure μ on $[0,1]^n$ and $c \in \mathbb{Q}^n_{\geqslant 0}$, the problem of computing an optimal Pareto cover of size k with respect to μ and c admits an FPTAS.*

We further show how to extend our approach to general product measures that satisfy some mild assumptions. Essentially, we will consider products of *nice* measures $(\mu_i)_{i=1}^n$ on $[0,1]$ that allow us to efficiently query an approximation of $\mu_i((a,b])$ for each a, b, i as well as a positive lower bound on the expectation of the identity on $[0,1]$ with respect to each μ_i. A more formal definition will be given later.

Our paper is structured as follows. In Section 2 we briefly discuss related work. The proofs of Theorem 1 and Proposition 2 are given in Section 3, where we also derive results for general discrete product measures that will be used in our FPTAS. The latter and hence the proof of Theorem 3 is presented in Section 4. The full version of this paper shows how to extend our FPTAS result to very general (product) measures. We close with some open questions in Section 5.

2 Related Work

To the best of our knowledge our setting for the general case $k > 2$ has not been studied in the literature. For the case $k = 2$ similar problems have been studied in the area of *stochastic optimization* in the context of *chance constrained optimization*. Here one aims to find an optimal solution to a problem with stochastic constraints. A solution to the problem then needs to fulfill the constraints with probability $1 - \delta$ for some $\delta > 0$.

Linear chance constrained problems are of the form

$$\min\{\langle c, x\rangle : x \in X, \mathbb{P}_{\xi \sim \mu}[Ax \geqslant \xi] \geqslant 1 - \delta\} \tag{2}$$

for a domain X, a distribution μ, matrix A and parameter δ. Note that our problem for $k = 2$ with linear cost functions f can be formulated as such a problem under the assumption that every Pareto cover has to contain $\mathbf{1}$. It remains to find the second vector in the optimal Pareto cover, for which one can guess the probability that it covers an element drawn from μ. This fits exactly into the framework of (2) where A is the identity matrix and $X = [0, 1]^n$. For an overview on the topic, we refer to the work of Nemirovski and Shapiro [4] and Luedtke, Ahmed, and Nemhauser [3]. In principle, it is possible to extend this idea to the case $k \geqslant 3$, for instance by using techniques from [3]. However, it is unclear whether theoretically efficient (approximation) algorithms can be obtained by such an approach.

The Pareto cover problem has an interesting interpretation in the context of tropical geometry. It can be equivalently phrased as a problem of partitioning $[0, 1]^n$ into k regions R_i. For each region, one selects the *tropical barycenter*, which is the coordinate-wise maximum of all its elements. This resembles a tropicalized version of classical Euclidean clustering algorithms and barycenters.

For randomized algorithms, tools from *statistical learning theory* (sample complexity, coresets, ...) can be used to study the Pareto cover problem. A natural approach would be to replace μ by a probability distribution $\tilde{\mu}$ that is a combination of polynomially many Dirac-distributions such that $\tilde{\mu}$ is "close" to μ, and reduce the problem to finding an optimal Pareto cover for $\tilde{\mu}$. In this work, however, we focus on deterministic algorithms and defer a discussion of such techniques to the full version of our paper.

3 Hardness results

In this section, we show that finding optimal Pareto covers (of given sizes) is a computationally hard problem, even for simple binary product measures and linear cost functions. In particular, we prove Theorem 1 and Proposition 2.

In the first part, we focus on the case where μ is a binary product measure given by some input vector $p \in [0, 1]^n$ such that $\mu = \mu_p$, see (1). We consider the following problem.

▶ **Definition 4.** The decision variant of the *binary Pareto cover problem* is defined as follows: Given $p \in ([0, 1] \cap \mathbb{Q})^n$, $c \in \mathbb{Q}^n_{\geqslant 0}$, $\gamma \in \mathbb{Q}$, and $k \in \mathbb{Z}_{\geqslant 1}$, decide whether there is some Pareto cover B of μ_p such that $|B| = k$ and $\mathrm{E}_{\mu_p}[c_B] \leqslant \gamma$.

We assume that p, c, and γ are given by their binary encodings. We leave open how k is encoded since in our applications it will be always polynomially bounded in n (see Proposition 2) or mostly be even a constant.

In Section 3.1, we show that the binary Pareto cover problem is weakly NP-hard if k is a fixed constant. In addition, we show that for $k = \frac{n+5}{3}$, the problem is strongly NP-hard. For the case that k is part of the input, we prove Proposition 2 showing that the problem of computing $\mathrm{E}_{\mu_p}[c_B]$ for a given Pareto cover B is #P-hard in Section 3.2. In view of this, it is unclear whether the binary Pareto cover problem is in NP if k is part of the input. However, for constant k, we establish in Section 3.3 that the problem is in NP, even for general discrete product measures. This result completes the proof of Theorem 1 and plays an important role in the design of our approximation algorithm.

3.1 NP-hardness

In this section, we consider the binary Pareto cover problem. Note that it can be solved efficiently in the case $k = 1$ since then $B = \{b\}$ is an optimal solution, where $b \in \{0, 1\}^n$ with $b_i = 1$ if and only if $p_i > 0$. In the remainder, we show that the binary Pareto cover problem is weakly NP-hard for $k = 2$. Similar, but slightly different proofs for the cases $k = 3$ and $k \geqslant 4$ can be found in the full version of this paper. Moreover, our reduction for $k \geqslant 4$ proves strong NP-hardness for $k = \frac{n+5}{3}$.

In order to show that the binary Pareto cover problem is NP-hard for $k = 2$, let us recall the `PARTITION` problem [2], which is well-known to be (weakly) NP-hard: Given $a_1, \dots, a_n \in \mathbb{Z}_{\geqslant 1}$ with $\sum_{i=1}^n a_i$ even, decide whether there is a subset $I \subseteq [n]$ such that $\sum_{i \in I} a_i = \frac{1}{2} \sum_{i=1}^n a_i$. We provide a reduction from `PARTITION` to the binary Pareto cover problem with $k = 2$.

Before defining it precisely, let us describe the idea first. Given a `PARTITION` instance $a_1, \dots, a_n$, set $\alpha := \frac{2}{a_1 + \dots + a_n}$ and consider an instance of the binary Pareto cover problem with $p_i = 1 - \mathrm{e}^{-\alpha \cdot a_i}$ and $c_i = a_i$ for all $i \in [n]$. Since all c_i and p_i are positive, every optimal Pareto cover of μ_p is of the form $B = \{b, \mathbf{1}\}$ where $b \in \{0,1\}^n$. Setting $I := \{i \in [n] : b_i = 0\}$, we see that the cost $\mathrm{E}_{\mu_p}[c_B]$ of B satisfies

$$\begin{aligned}\sum_{i=1}^n a_i - \mathrm{E}_{\mu_p}[c_B] &= \mathbf{1}^\intercal c - \mathrm{E}_{\mu_p}[c_B] \\ &= \mathbf{1}^\intercal c - c^\intercal b \cdot \mu_p(\{x \in \{0,1\}^n : x \leqslant b\}) - c^\intercal \mathbf{1} \cdot \mu_p(\{x \in \{0,1\}^n : x \not\leqslant b\}) \\ &= c^\intercal (\mathbf{1} - b) \cdot \mu_p(\{x \in \{0,1\}^n : x \leqslant b\}) \\ &= \prod\nolimits_{i \in I} (1 - p_i) \cdot \sum\nolimits_{i \in I} a_i \\ &= \mathrm{e}^{-\alpha \sum_{i \in I} a_i} \cdot \sum\nolimits_{i \in I} a_i \\ &= h\left(\sum\nolimits_{i \in I} a_i\right),\end{aligned}$$

where $h \colon \mathbb{R} \to \mathbb{R}$, $h(x) = x \cdot \mathrm{e}^{-\alpha \cdot x}$. Since h has its unique maximum at $x = \alpha^{-1}$, we see that $B = \{b, \mathbf{1}\}$ is a Pareto cover with $\mathrm{E}_{\mu_p}[c_B] \leqslant \sum_{i=1}^n a_i - h(\alpha^{-1})$ if and only if $I = \{i \in [n] : b_i = 0\}$ satisfies $\sum_{i \in I} a_i = \alpha^{-1} = \frac{1}{2} \sum_{i=1}^n a_i$. In other words, if $a_1, \dots, a_n$ is a "yes" instance for `PARTITION`, then there is a Pareto B cover of size $k = 2$ of μ_p with cost $\mathrm{E}_{\mu_p}[c_B] \leqslant \sum_{i=1}^n a_i - h(\frac{1}{2} \sum_{i=1}^n a_i)$. If $a_1, \dots, a_n$ is a "no" instance, then every Pareto cover of size $k = 2$ will have cost $\mathrm{E}_{\mu_p}[c_B] > \sum_{i=1}^n a_i - h(\frac{1}{2} \sum_{i=1}^n a_i)$.

Unfortunately, the probabilities that we have used in the above argument cannot be polynomially represented. However, we show that we can efficiently round them such that the above strategy still works. More specifically, our probabilities and the threshold cost γ will be defined as follows.

▶ **Lemma 5.** *Given $a_1, \dots, a_n \in \mathbb{Z}_{\geqslant 1}$, we can compute $p_1, \dots, p_n \in [0,1]$ and $\gamma \in \mathbb{Q}$ in polynomial time such that*

$$\frac{1-\beta}{\mathrm{e}^{\alpha \cdot a_i}} \leqslant 1 - p_i \leqslant \frac{1+\beta}{\mathrm{e}^{\alpha \cdot a_i}} \quad \text{for all } i \in [n], \text{ and} \quad \frac{(1-\beta)^{n+2}}{\alpha \cdot \mathrm{e}} \leqslant \sum_{i=1}^n a_i - \gamma \leqslant \frac{(1-\beta)^n}{\alpha \cdot \mathrm{e}},$$

where $\alpha := \frac{2}{\sum_{i=1}^n a_i}$ and $\beta := \frac{\alpha^2}{48(n+1)}$.

Let $p_1, \dots, p_n$ and γ be given as in the above statement. Note that we have $0 < p_i \leqslant 1$ for all $i \in [n]$ since $p_i \leqslant 1 - \frac{1-\beta}{\mathrm{e}^{\alpha \cdot a_i}} \leqslant 1$ and

$$p_i \geqslant 1 - \frac{1+\beta}{\mathrm{e}^{\alpha \cdot a_i}} > 1 - \frac{1+\beta}{1 + \alpha \cdot a_i} \geqslant 1 - \frac{1+\beta}{1+\alpha} > 0.$$

Finally, set $c_i := a_i$ for all $i \in [n]$. It remains to prove the following lemma.

▶ **Lemma 6.** *There is a subset $I \subseteq [n]$ with $\sum_{i\in I} a_i = \frac{1}{2}\sum_{i=1}^n a_i$ if and only if there is a Pareto cover B of μ_p with $|B| = 2$ and $\mathrm{E}_{\mu_p}[c_B] \leqslant \gamma$.*

The proves of the above lemmas can be found in the full version.

Note that we have shown that the binary Pareto cover problem is weakly NP-hard for constant k. We remark that, unless $P = NP$, the problem cannot be strongly NP-hard, as we derive an FPTAS in Section 4.

3.2 #P-hardness

In the previous section we have seen that the binary Pareto cover problem is NP-hard, already for constant k. The next natural question is whether the problem is in NP. At first sight, a Pareto cover $B \subseteq \{0,1\}^n$ itself seems to be a canonical certificate for a "yes" instance. However, with this choice, we should be able to compute the cost $\mathrm{E}_{\mu_p}[c_B]$ efficiently. Unfortunately, if k is not part of the input, i.e., B is not of constant size, computing $\mathrm{E}_{\mu_p}[c_B]$ is hard. More precisely, let us prove Proposition 2, which states that, given a Pareto cover B for the uniform distribution μ on $\{0,1\}^n$, the problem of computing $\mathrm{E}_\mu[\mathbf{1}_B]$ is #P-hard.

Proof of Proposition 2. We use the fact that the problem of computing the number of vertex covers in a given undirected graph is #P-hard [5]. Given a graph $G = (V, E)$, identify V with $[n]$ and for every edge $e \in E$, let b_e denote the characteristic vector of $V \setminus e$, the set of nodes that are not part of e. Let μ denote the uniform distribution on $\{0,1\}^n$, i.e., $\mu = \mu_p$ with $p = \frac{1}{2} \cdot \mathbf{1}$. Consider the Pareto cover $B := \{b_e : e \in E\} \cup \{\mathbf{1}\}$. Setting $c = \mathbf{1}$, the cost of B is equal to

$$\begin{aligned}
\mathrm{E}_\mu[c_B] &= (n-2) \cdot \mu(\{x \in \{0,1\}^n : x \leqslant b_e \text{ for some } e \in E\}) \\
&\qquad + n \cdot \mu(\{x \in \{0,1\}^n : x \nleqslant b_e \text{ for all } e \in E\}) \\
&= (n-2) + 2 \cdot \mu(\{x \in \{0,1\}^n : x \nleqslant b_e \text{ for all } e \in E\}) \\
&= (n-2) + \frac{|\{x \in \{0,1\}^n : x \nleqslant b_e \text{ for all } e \in E\}|}{2^{n-1}} \\
&= (n-2) + \frac{|\{U \subseteq [n] : U \cap e \neq \emptyset \text{ for all } e \in E\}|}{2^{n-1}},
\end{aligned}$$

and hence we see that $2^{n-1} \cdot (\mathrm{E}_\mu[c_B] - (n-2))$ is the number of vertex covers in G. ◀

3.3 Membership in NP for constant k

For the binary Pareto cover problem, we have seen that computing the cost of a given Pareto cover B is hard if B can be of any size. In this section, we show that the cost can be computed efficiently if B is of constant size. In fact, we prove that this is the case for the discrete version of our problem (see (1)).

We introduce the following notation: For probability measures $(\mu_i)_{i=1}^n$ defined on (the Borel σ-algebra on) $[0,1]$, we define their product $\mu := \prod_{i=1}^n \mu_i$ to be given by $\mu(I_1 \times \cdots \times I_n) = \prod_{i=1}^n \mu_i(I_i)$, where $(I_i)_{i=1}^n$ are intervals contained in $[0,1]$. In particular, for $A = \{a_0, \dots, a_{M+1}\}$ and $p = \left((p_j^i)_{j=0}^{M+1}\right)_{i=1}^n$ as in (1), $\mu = \mu_{A,p}$ is the product of the measures $\mu_i = \mu_{i,A,p}$ given by $\mu_i(\{a_j\}) = p_j^i$, $i \in [n]$, $j = 0, \dots, M+1$. We study the following problem.

▶ **Definition 7.** The decision variant of the *discrete Pareto cover problem* is the following: Given $A = \{a_0, \dots, a_{M+1}\}$ with $0 = a_0 < a_1 < \cdots < a_M < a_{M+1} = 1$, $p = \left((p_j^i)_{j=0}^{M+1}\right)_{i=1}^n$ with $p_j^i \in [0,1]$ and $\sum_{j=0}^{M+1} p_j^i = 1$ for all i, $c \in \mathbb{Q}_{\geqslant 0}^n$, $\gamma \in \mathbb{Q}$, and $k \in \mathbb{Z}_{\geqslant 1}$, decide whether there is some Pareto cover B of $\mu_{A,p}$ such that $|B| = k$ and $\mathrm{E}_{\mu_p}[c_B] \leqslant \gamma$.

Again, we assume that A, p, c, and γ are given by their binary encodings. In our applications, k will be always constant.

Note that it is easy to check whether a finite set $B \subseteq [0,1]^n$ is feasible for the above problem, i.e., that it is a Pareto cover of $\mu_{A,p}$. In fact, let $x^* \in [0,1]^n$ be given by $x^*_i := \max\{a_j : p^i_j > 0\}$ for $i \in [n]$. Then B is a Pareto cover of $\mu_{A,p}$ if and only if $|B| = k$ and B contains at least one point that covers x^*.

Moreover, note that if B is feasible for the above problem, then we may assume that $B \subseteq A^n$ holds since otherwise we may lower entries of points in B without changing the set of points they cover and without increasing their cost.

▶ **Proposition 8.** *Let $k \in \mathbb{Z}_{\geqslant 1}$ be fixed. Given A, p, c as in the discrete Pareto cover problem and a Pareto cover $B \subseteq A^n$ for $\mu = \mu_{A,p}$ with $|B| = k$, we can compute $\mathrm{E}_\mu[c_B]$ in polynomial time.*

Note that this shows that the discrete Pareto cover problem (and hence also the binary Pareto cover problem) is in NP if k is constant.

In order to prove Proposition 8, we make use of the following notation. For $i \in [n]$ vectors $b^1, \ldots, b^k, x \in [0,1]^n$, we define $J^i(x)$ to consist of all indices $j \in [k]$ such that b^j covers x, if we restrict both vectors to the first i coordinates. More precisely, we set

$$J^i(x) := \left\{ j \in [k] : x_1 \leqslant b^j_1, \ldots, x_i \leqslant b^j_i \right\} \text{ for } i \in [n], \text{ and let } J(x) := J^n(x).$$

Whenever we refer to $J(x), J^1(x), \ldots, J^n(x)$, the vectors $b^1, \ldots, b^k$ will be clear from the context. Observe that for $i \in [n]$ and $J \subseteq [n]$, the set $\{x \in [0,1]^n : J^i(x) = J\}$ is a Borel set.

Note that $\{b^1, \ldots, b^k\}$ is a Pareto cover for μ if and only if $\mu(\{x \in [0,1]^n : J(x) = \emptyset\}) = 0$. Let us rephrase the cost of a Pareto cover using this new notation:

▶ **Lemma 9.** *Let μ be a probability measure on (the Borel σ-algebra on) $[0,1]^n$ and let $B = \{b^1, \ldots, b^k\}$ be a Pareto cover of μ. Then for every $c \in \mathbb{R}^n$ we have*

$$\mathrm{E}_\mu[c_B] = \sum_{\emptyset \neq J \subseteq [k]} \mu(\{x \in [0,1]^n : J(x) = J\}) \cdot \min_{j \in J} c^\intercal b^j.$$

Proof. For $J \subseteq [k]$, note that $c_B(x) = \min_{j \in J} c^\intercal b^j$ holds for all $x \in [0,1]^n$ with $J(x) = J$. The claim follows since the sets $(\{x \in [0,1]^n : J(x) = J\})_{J \subseteq [k]}$ are disjoint. ◀

Thus, in order to prove Proposition 8, it suffices to show that we can compute the values $\mu_{A,p}(\{x \in [0,1]^n : J(x) = J\})$ for all $J \subseteq [k]$ in polynomial time. To this end, we show how to iteratively compute the values $\mu_{A,p}(\{x \in [0,1]^n : J^i(x) = J\})$ for all $J \subseteq [k]$ and $i \in [n]$. Lemma 10 takes care of the base case $i = 1$, whereas Lemma 11 explains how to proceed from i to $i + 1$.

▶ **Lemma 10.** *Let $\mu_1, \ldots, \mu_n$ be probability measures on $[0,1]$ and let $b^1, \ldots, b^k \in [0,1]^n$. For $\mu = \prod_{i=1}^n \mu_i$ and $J \subseteq [k]$ we have*

$$\mu(\{x \in [0,1]^n : J^1(x) = J\}) = \mu_1 \left((\alpha, \beta] \cap [0,1]\right),$$

where $\alpha = \max_{j \in [k] \setminus J} b^j_1$ and $\beta = \min_{j \in J} b^j_1$.

Here, we use the convention $\max \emptyset := -\infty$ and $\min \emptyset := +\infty$.

Proof of Lemma 10. We have $J^1(x) = J$ if and only if $x_1 \leqslant b^j_1$ for all $j \in J$ and $x_1 > b^j_1$ for all $j \in [k] \setminus J$. That is, $\{x \in [0,1]^n : J^1(x) = J\} = \{x \in [0,1]^n : x_1 \in (\alpha, \beta]\}$. ◀

▶ **Lemma 11.** *Let $\mu_1, \dots, \mu_n$ be probability measures on $[0,1]$ and let $b^1, \dots, b^k \in [0,1]^n$. For $\mu = \prod_{i=1}^n \mu_i$, $J \subseteq [k]$, and $i \in \{2, \dots, n\}$ we have*

$$\mu(\{x \in [0,1]^n : J^i(x) = J\}) = \sum_{J \subseteq L \subseteq [k]} \mu(\{x \in [0,1]^n : J^{i-1}(x) = L\}) \cdot \mu_i((\alpha_L, \beta_L] \cap [0,1]),$$

where $\alpha_L = \max_{j \in L \setminus J} b_i^j$ and $\beta_L = \min_{j \in J} b_i^j$.

Proof. The claim follows from the fact that $\{x \in [0,1]^n : J^i(x) = J\}$ is equal to

$$\bigcup_{J \subseteq L \subseteq [k]} \Big[\{x \in [0,1]^n : J^{i-1}(x) = L\} \\ \cap \{x \in [0,1]^n : x_i \leqslant b_i^j \text{ for all } j \in J,\ x_i > b_i^j \text{ for all } j \in L \setminus J\}\Big]$$

and the observation that the above sets are disjoint. ◀

Note that for measures $\mu = \mu_{A,p} = \prod_{i=1}^n \mu_i$ and $\alpha, \beta \in \mathbb{Q} \cup \{\pm\infty\}$, we can compute $\mu_i((\alpha, \beta] \cap [0,1])$ in polynomial time. Moreover, for constant k, the sum in Lemma 11 only has a constant number of summands. This yields Proposition 8.

4 Approximation algorithm

The goal of this section is to develop an FPTAS (see [1]) for the discrete Pareto cover problem, see Theorem 3. More precisely, we provide an algorithm that receives an instance I of the discrete Pareto cover problem and a parameter $\gamma \in (0,1) \cap \mathbb{Q}$, and computes a $(1+\gamma)$-approximate solution to I in time polynomial in γ^{-1} and the encoding length of I. All proofs and an extension of our FPTAS to more general product measures can be found in the full version.

Let A, p, c, k define an instance of the discrete Pareto cover problem, where k is a constant. For every $i \in [n]$ let $l_i := \max\{l : p_l^i > 0\}$ and define $a^* := (a_{l_1}, \dots, a_{l_n})$. Recall that every Pareto cover B of $\mu = \mu_{A,p}$ must contain a point that covers a^*. Conversely, every finite set $B \subseteq [0,1]^n$ containing a^* is a Pareto cover of μ. Since the costs are non-negative, we can restrict ourselves to Pareto covers that contain a^*.

Next, we discuss how to determine a cover of approximately minimum cost. Recall that in Section 3.3, Lemma 9, given a Pareto cover $B = \{b^1, \dots, b^k\}$, we have seen that we can express our objective as

$$\mathrm{E}_\mu[c_B] = \sum_{\emptyset \neq J \subseteq [k]} \mu(\{x \in [0,1]^n : J(x) = J\}) \cdot \min_{j \in J} c(b^j).$$

Even more, we know that we can iteratively compute the values $\mu(\{x \in [0,1]^n : J^i(x) = J\})$ for $i \in [n]$ and $J \subseteq [k]$ in polynomial time, see Lemma 11. In doing so, the only information we need to proceed from i to $i+1$ are the probabilities $\mu(\{x \in [0,1]^n : J^i(x) = J\})$ for $J \subseteq [k]$ and the values $(b_{i+1}^j)_{j=1}^k$, but *no* further information on the values b_l^j for $l \in [i]$ and $j \in [k]$. What is more, by definition, the values $\mu(\{x \in [0,1]^n : J^i(x) = J\})$ for $J \subseteq [k]$ and $\sum_{l=1}^i c_l \cdot b_l^j$ for $j \in [k]$ do not depend on the coordinates b_l^j, $l = i+1, \dots, k$, $j \in [k]$. All in all, these are the best preconditions for a dynamic programming approach and motivate the following definition:

▶ **Definition 12.** For a Pareto cover $B = (b^j)_{j=1}^k$ with $b^k = a^*$ and $i \in [n]$ let

$$\mathrm{Cand}^i(B) := (i, (P_J)_{J \subseteq [k]}, (C_j)_{j=1}^k),$$

where $P_J = \mu(\{x \in [0,1]^n : J^i(x) = J\})$ for $J \subseteq [k]$ and $C_j = \sum_{l=1}^i c_l \cdot b_l^j$ for $j \in [k]$.

▶ **Definition 13.** A *candidate* is a triple $\mathcal{C} = (i, (P_J)_{J\subseteq[k]}, (C_j)_{j=1}^k)$.

The *cost* of the candidate $\mathcal{C}$ is given by $\mathrm{Cost}(\mathcal{C}) := \sum_{\emptyset\neq J\subseteq[k]} P_J \cdot \min_{j\in J} C_j$.

We call $\mathcal{C}$ *valid* if there exists a Pareto cover $B = (b^j)_{j=1}^k$ with $b^k = a^*$ such that $\mathcal{C} = \mathrm{Cand}^i(B)$, and say that B *witnesses* the validity of the candidate.

Note that the definition of the cost of a candidate is in accordance with Lemma 9.

A naive approach to tackle the discrete Pareto cover problem would now be to iteratively enumerate all valid candidates for $i = 1, \dots, n$, select a candidate for $i = n$ that yields the minimum objective value, and then back-trace to compute a corresponding cover. The problem with this idea is of course that we do not have a polynomial bound on the number of candidates we generate. To overcome this issue, we round the candidates appropriately to ensure a polynomial number of possible configurations, whilst staying close enough to the original values to obtain a good approximation of the objective for $i = n$. Observe that for constant k, the number of entries of each candidate is constant, which means that it suffices to polynomially bound the number of values each of them may attain.

To this end, consider Algorithm 1. The gray lines are not part of the algorithm itself, but only needed for its analysis. Before diving into the analysis of Algorithm 1, we would like to provide some intuition about what is happening. We start by enumerating all possible values $(b_1^j)_{j=1}^k$ may attain in a solution B with $b^k = a^*$ and use this information to compute $\mathrm{Cand}^1(B)$ according to Definition 12. (Recall that this is independent of the values b_l^j for $l \geqslant 2$, $j = 1, \dots, k$.) Then, we round all non-zero entries of $\mathrm{Cand}^1(B)$ (except for the first one, which is 1) down to the next power of $1 + \delta$, $\delta = \frac{\epsilon}{4n}$. Each rounded candidate $\mathcal{C}$ is added to our table $\mathcal{T}$, and for back-tracing purposes, we store a cover B that leads to $\mathcal{C}$ as $\mathrm{Witness}(\mathcal{C})$. For the analysis, we further maintain an imaginary map AllWits mapping each rounded candidate $\mathcal{C} \in \mathcal{T}$ to the set of all possible witness covers that result in $\mathcal{C}$ after (iterative) rounding.

After dealing with the base case $i = 1$, we enumerate possible values of $(b_i^j)_{j=1}^k$ for $i = 2, \dots, n$, loop over all rounded candidates for $i - 1$ and compute new rounded candidates for i according to Lemma 11. Moreover, we deduce witnesses for our new candidates for i from those stored for the candidates for $i - 1$ and the values $(b_i^j)_{j=1}^k$. This might of course lead to an exponential growth of the size of the imaginary map AllWits. However, the fact that the Witness-map only memorizes one witness per candidate keeps the total running time under control, provided we can come up with a polynomial bound on the number of candidates we generate. Lemma 14 takes care of this, and is the main ingredient of the proof of Theorem 15, which guarantees a polynomial running time.

▶ **Lemma 14.** *At each point during the algorithm, we have $|\mathcal{T}| \leqslant \alpha^{2^k} \cdot \beta^k \cdot n$, where*

$$\alpha = n + 2 - n \cdot \min\left\{\log_{1+\delta}(p_l^i) : i \in [n],\, l \in \{0, \dots, M+1\},\, p_l^i > 0\right\},$$
$$\beta = n + 2 + \log_{1+\delta}(c_1 + \cdots + c_n) - \log_{1+\delta}(a_1) - \min\{\log_{1+\delta}(c_i) : i \in [n]\}.$$

In particular, for constant k, $|\mathcal{T}|$ is polynomially bounded in the encoding length of the given instance of the discrete Pareto cover problem and ϵ^{-1}.

▶ **Theorem 15.** *Given an instance I of the discrete Pareto cover problem and a parameter $\epsilon \in (0,1) \cap \mathbb{Q}$ as input, Algorithm 1 runs in time polynomial in* $\mathrm{size}(I)$ *and ϵ^{-1}.*

Denote the set of all candidate in $\mathcal{T}$ starting with i by $\mathcal{T}_i$. In order to finally obtain an FPTAS for the discrete Pareto cover problem, our goal for the remainder of this section is to prove the following result:

▶ **Theorem 16.** *Running Algorithm 1 and choosing the witness of a candidate of minimum cost in $\mathcal{T}_n$ yields a $(1 + \epsilon)$-approximation.*

Algorithm 1 Dynamic program to compute rounded candidates.

Input: $(a_l)_{l=0}^{M+1}$, $((p_l^i)_{l=0}^{M+1})_{i=1}^n$, $(c_i)_{i=1}^n$, k, $\epsilon \in \mathbb{Q} \cap (0,1)$
Output: a table $\mathcal{T}$ of rounded candidates

For $i = 1, \dots, n$ compute $l_i := \max\{l : p_l^i > 0\}$.
$a^* \leftarrow (a_{l_i})_{i=1}^n$, $A \leftarrow \{a_0, \dots, a_{M+1}\}$,$\mathcal{T} \leftarrow \emptyset$
$\mathrm{AllWits}(-) \leftarrow \emptyset$
$\delta \leftarrow \frac{\epsilon}{4n}$
foreach $(\beta_j)_{j=1}^k \in A^k$ *with* $\beta_k = a_1^*$ **do**
  Define $(P_J)_{J\subseteq[k]}$ by $P_J \leftarrow \mu_1((\max_{j\in[k]\setminus J}\beta_j, \min_{j\in J}\beta_j] \cap [0,1])$
  $P_J \leftarrow \begin{cases} (1+\delta)^{\lfloor \log_{1+\delta} P_J \rfloor} & , P_J > 0 \\ 0 & , P_J = 0 \end{cases}$
  Define $(C_j)_{j=1}^k$ by $C_j \leftarrow \begin{cases} (1+\delta)^{\lfloor \log_{1+\delta}(c_1 \cdot \beta_j) \rfloor} & , c_1 \cdot \beta_j > 0 \\ 0 & , c_1 \cdot \beta_j = 0 \end{cases}$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{(1, (P_J)_{J\subseteq[k]}, (C_j)_{j=1}^k)\}$
  $b_1^j \leftarrow \beta_j$, $j = 1, \dots, k$, $b_i^j \leftarrow 0$, $i = 2, \dots, n$, $j = 1, \dots, k-1$, $b_i^k \leftarrow a_i^*$, $i = 2, \dots, n$
  $\mathrm{Witness}((1, (P_J)_{J\subseteq[k]}, (C_j)_{j=1}^k)) \leftarrow (b^j)_{j=1}^k$
  $\mathrm{AllWits}((1, (P_J)_{J\subseteq[k]}, (C_j)_{j=1}^k)) \leftarrow \mathrm{AllWits}((1, (P_J)_{J\subseteq[k]}, (C_j)_{j=1}^k)) \cup \{(b^j)_{j=1}^k\}$
for $i = 2$ *to* n **do**
  foreach $(i-1, (P_J^{i-1})_{J\subseteq[k]}, (C_j^{i-1})_{j=1}^k) \in \mathcal{T}$ **do**
    foreach $(\beta_j)_{j=1}^k \in A^k$ *with* $\beta_k = a_i^*$ **do**
      Define $(P_J^i)_{J\subseteq[k]}$ by
      $P_J^i \leftarrow \sum_{\substack{L: \\ J\subseteq L\subseteq[k]}} P_L^{i-1} \cdot \mu_i((\max_{j\in L\setminus J}\beta_j, \min_{j\in J}\beta_j] \cap [0,1])$
      $P_J^i \leftarrow \begin{cases} (1+\delta)^{\lfloor \log_{1+\delta} P_J^i \rfloor} & , P_J^i > 0 \\ 0 & , P_J^i = 0 \end{cases}$
      Define $(C_j^i)_{j=1}^k$ by $C_j^i \leftarrow C_j^{i-1} + c_i \cdot \beta_j$
      $C_j^i \leftarrow \begin{cases} (1+\delta)^{\lfloor \log_{1+\delta} C_j^i \rfloor} & , C_j^i > 0 \\ 0 & , C_j^i = 0 \end{cases}$
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{(i, (P_J^i)_{J\subseteq[k]}, (C_j^i)_{j=1}^k)\}$
      $(b^{i-1,j})_{j=1}^k \leftarrow \mathrm{Witness}((i-1, (P_J^{i-1})_{J\subseteq[k]}, (C_j^{i-1})_{j=1}^k))$
      Define $(b^{i,j})_{j=1}^k$ by $b_l^{i,j} := \begin{cases} b_l^{i-1,j} & , l \neq i \\ \beta_j & , l = i \end{cases}$
      $\mathrm{Witness}((i, (P_J^i)_{J\subseteq[k]}, (C_j^i)_{j=1}^k)) \leftarrow (b^{i,j})_{j=1}^k$
      foreach $(\tilde{b}^{i-1,j})_{j=1}^k \in \mathrm{AllWits}((i-1, (P_J^{i-1})_{J\subseteq[k]}, (C_j^{i-1})_{j=1}^k))$ **do**
        Define $(\tilde{b}^{i,j})_{j=1}^k$ by $\tilde{b}_l^{i,j} := \begin{cases} \tilde{b}_l^{i-1,j} & , l \neq i \\ \beta_j & , l = i \end{cases}$
        $\mathrm{AllWits}((i, (P_J^i)_{J\subseteq[k]}, (C_j^i)_{j=1}^k)) \leftarrow$
        $\mathrm{AllWits}((i, (P_J^i)_{J\subseteq[k]}, (C_j^i)_{j=1}^k)) \cup \{(\tilde{b}^{i,j})_{j=1}^k\}$
return $\mathcal{T}$

The proof of Theorem 16 consists of two main steps. Lemma 17 shows that any rounded candidate $\tilde{\mathcal{C}}$ we store in $\mathcal{T}$ invokes similar costs to those of any of its witness covers. Proposition 18 ensures that every cover $B = (b^j)_{j=1}^k$ with $b^k = a^*$ occurs as a possible witness for some candidate. In particular, this holds for an optimum cover B^* (with $b^{*,k} = a^*$) and by Lemma 17, we can therefore infer that the costs of the solution we return can only be by a factor of $1+\epsilon$ larger than the optimum.

▶ **Lemma 17.** *Let $\tilde{\mathcal{C}} \in \mathcal{T}_n$ and let $B \in \mathrm{AllWits}(\tilde{\mathcal{C}})$. Then $\mathrm{Cost}(\tilde{\mathcal{C}}) \leqslant \mathrm{E}_\mu[c_B] \leqslant (1+\epsilon)\cdot\mathrm{Cost}(\tilde{\mathcal{C}})$.*

▶ **Proposition 18.** *Let an instance of the discrete Pareto cover problem be given. For each $i \in [n]$ and each cover $B = (b^j)_{j=1}^k$ such that $b^k = a^*$ and $b^j_l = 0$ for $j = 1\dots,k-1$ and $l = i+1,\dots,n$, there exists $\mathcal{C} \in \mathcal{T}_i$ such that $B \in \mathrm{AllWits}(\mathcal{C})$.*

Combining Theorem 16, Lemma 14 and Theorem 15 and observing that we can compute the cost of a candidate in polynomial time for constant k, we obtain Theorem 3, which we restate once again:

▶ **Theorem 3.** *Let $k \in \mathbb{N}$ be fixed. Given a discrete product measure μ on $[0,1]^n$ and $c \in \mathbb{Q}^n_{\geqslant 0}$, the problem of computing an optimal Pareto cover of size k with respect to μ and c admits an FPTAS.*

5 Conclusion

In this paper, we have introduced the Pareto cover problem and studied the case of product measures and linear cost functions. For fixed k, we have come to a pretty good understanding of its complexity: On the one hand, we could show weak NP-hardness of the problem. On the other hand, we have established the existence of an FPTAS.

However, there are several questions that remain open and constitute an interesting subject for future research. To begin with, we have seen that even in a very restricted setting such as the uniform probability distribution on $[0,1]^n$, it seems non-trivial to find an optimum cover (see Figure 1). Consequently, in order to obtain a better feeling for the problem at hand, it can be worthwhile to examine the structure of optimum solutions for such special cases.

When dealing with the binary problem variant, another question that comes up is for which subsets of $\{0,1\}^n$, there exists an instance they are optimum for. Any insights towards this question may lead to new, perhaps more efficient strategies to tackle the Pareto cover problem.

In addition to these rather concrete questions, there are also several more fundamental issues one may want to address. For instance, even though we have seen that computing the objective value attained by an arbitrary solution is #P-hard in general (i.e., for non-fixed k), this does not resolve containment in NP since there might be another choice of a certificate that does the trick. More generally, it would be interesting to fully understand the dependence of the problem complexity on the parameter k. To this end, note that the complexity does not simply "increase" with larger values of k, given that in the discrete setting, the problem is weakly NP-hard for constant $k \geqslant 2$, and strongly NP-hard, e.g., for $k = \frac{n+5}{3}$, but once k is at least as large as the number of all possible discrete vectors b, it is obvious what an optimum solution should look like (and we can output it in polynomial time, assuming an appropriate output encoding is chosen). Hence, it seems interesting to further investigate the hardness transition of the problem: When exactly does the problem become strongly NP-hard? When does it become easier again?

Finally, as all of our results apply to the case of product measures, it appears natural to ask what can be done for general probability measures. To this end, as alluded to in Section 2, it could be fruitful to explore connections to existing results from statistical learning theory.

References

1. Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer Science & Business Media, 1999. doi:10.1007/978-3-642-58412-1.
2. Michael R Garey and David S Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990. URL: https://dl.acm.org/doi/10.5555/574848.
3. James Luedtke, Shabbir Ahmed, and George L Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical programming*, 122(2):247–272, 2010. doi:10.1007/s10107-008-0247-4.
4. Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007. doi:10.1137/050622328.
5. Salil P Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM Journal on Computing*, 31(2):398–427, 2001. doi:10.1137/S0097539797321602.