

On the PTAS Complexity of Multidimensional Knapsack

Ilan Doron-Arad   

MIT, Cambridge, MA, USA

Ariel Kulik   

Ben-Gurion University of the Negev, Be'er-Sheva, Israel

Pasin Manurangsi   

Google Research, Bangkok, Thailand

Abstract

We study the d -dimensional knapsack problem. We are given a set of items, each with a d -dimensional cost vector and a profit, along with a d -dimensional budget vector. The goal is to select a set of items that do not exceed the budget in all dimensions and maximize the total profit. A polynomial-time approximation scheme (PTAS) with running time $n^{\Theta(d/\varepsilon)}$ has long been known for this problem, where ε is the error parameter and n is the encoding size. Despite decades of active research, the best running time of a PTAS has remained $O(n^{\lceil d/\varepsilon \rceil - d})$. Unfortunately, existing lower bounds only cover the special case with two dimensions $d = 2$, and do not answer whether there is a $n^{o(\frac{d}{\varepsilon})}$ -time PTAS for larger values of d .

In this work, we show that the running times of the best-known PTAS cannot be improved up to a polylogarithmic factor assuming the *Exponential Time Hypothesis (ETH)*. Our techniques are based on a robust reduction from 2-CSP, which *embeds* 2-CSP constraints into a desired number of dimensions. Then, using a recent result of [Bafna Karthik and Minzer, STOC'25], we succeed in exhibiting tight trade-off between d and ε for all regimes of the parameters assuming d is sufficiently large. Informally, our result also shows that under ETH, for any function f there is no $f(\frac{d}{\varepsilon}) \cdot n^{\tilde{O}(\frac{d}{\varepsilon})}$ -time $(1 - \varepsilon)$ -approximation for d -dimensional knapsack, where n is the number of items and $\tilde{O}$ hides polylogarithmic factors in $\frac{d}{\varepsilon}$.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Approximation algorithms analysis; Theory of computation $\rightarrow$ Problems, reductions and completeness; Theory of computation $\rightarrow$ Parameterized complexity and exact algorithms

Keywords and phrases d -dimensional Knapsack, Multidimensional Knapsack, PTAS, CSP

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.50

Related Version *Previous Version:* <https://arxiv.org/abs/2407.10146> [17]

Acknowledgements Pasin is grateful to Karthik C. S. for insightful discussions on [7].

1 Introduction

In the 0/1-Knapsack problem, we are given a set of items, each with a cost and a profit, together with a budget. The goal is to select a subset of items whose total cost does not exceed the budget while maximizing the total profit. Knapsack is one of the most studied problems in combinatorial optimization. Indeed, a *polynomial-time approximation scheme* (PTAS) for the problem [28, 48] was among the first PTASes to have ever been discovered in the field, and is often the first PTAS taught in computer science courses (see e.g. [50, Section 3.1] and [13, Chapter 35.5]). While such a PTAS has been known to exist since the 70s [28, 48], the problem remains an active topic of research to this day [23, 37, 32, 33, 46, 9, 27, 16, 42, 12], with the best known (fully) PTAS that gives a $(1 - \varepsilon)$ -approximation in time $\tilde{O}(n + 1/\varepsilon^2)$ [42, 12]. This running time is essentially tight assuming (min, +)-convolution cannot be solved in truly sub-quadratic time [15, 36].



© Ilan Doron-Arad, Ariel Kulik, and Pasin Manurangsi;
licensed under Creative Commons License CC-BY 4.0

17th Innovations in Theoretical Computer Science Conference (ITCS 2026).

Editor: Shubhangi Saraf; Article No. 50; pp. 50:1–50:22



Leibniz International Proceedings in Informatics

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

Numerous generalizations of Knapsack have been considered over the years. One of the most natural and well-studied versions is the so-called *d-dimensional knapsack* problem. Here each cost is a d -dimensional vector. The problem remains the same, except that the total cost must not exceed the budget in every coordinate. The problem is defined more formally below.

► **Definition 1** (*d*-Dimensional Knapsack).

Input: An instance $\mathcal{I} = (I, p, c, B)$ consisting of:

- a set of items I ,
- a profit function $p : I \rightarrow \mathbb{N}$,
- a d -dimensional cost function $c : I \rightarrow \mathbb{N}^d$,
- a d -dimensional budget $B \in \mathbb{N}^d$.

Solution: A subset $S \subseteq I$ such that for all $j \in [d]$ it holds that $c_j(S) := \sum_{i \in S} c_j(i) \leq B_j$.

Objective: Find a solution of maximum profit, where the profit of S is $p(S) := \sum_{i \in S} p(i)$.

Multidimensional (vector) knapsack is the family of d -dimensional knapsack problems for all $d \geq 1$. The study of d -dimensional knapsack dates back to the 50s [40, 43, 49] and it has been used to model problems arising in many contexts, including resource allocation, delivery system, budgeting, investment, and voting (see the survey [18] for a more comprehensive review). Given its importance, there have been several algorithms and hardness results devised for the problem over the years. We summarize the main theoretical results below. We note that many heuristics for the problems have been proposed (e.g., [2, 5, 6, 31, 38, 47, 3, 51, 45]), although these do not provide theoretical guarantees and thus will be omitted from the subsequent discussions.

On the hardness front, the decision version of the Knapsack problem ($d = 1$) generalizes the Subset Sum problem, which is one of Karp's 21 NP-complete problems [29]. The $d = 1$ case admits a *fully polynomial time approximation scheme* (FPTAS), i.e., a PTAS that runs in time $(n/\varepsilon)^{O(1)}$. In contrast, for $d = 2$, it has also long been known that no FPTAS exists unless $P = NP$ [34, 41]. This has more recently been strengthened in [35], who show that, even when $d = 2$, the problem does not admit an efficient PTAS (EPTAS), i.e., one that runs in time $f(\varepsilon) \cdot n^{O(1)}$ time, assuming $W[1] \neq \text{FPT}$. This in turn was improved by [26] to rule out any PTAS that runs in time $n^{o(1/\varepsilon)}$ even for $d = 2$, assuming ETH. Recently, [20] established a lower bound for exact algorithms for the d -Dimensional Knapsack problem, based on a hypothesis concerning multi-dimensional (max, +) convolutions.

Meanwhile, on the algorithmic front, a PTAS for d -dimensional knapsack with running time $n^{O(d/\varepsilon)}$ was first described by Chandra et al. [10]¹, not long after a PTAS for Knapsack was discovered in [28, 48]. The d -dimensional knapsack has been a challenging research topic ever since: over the past nearly 50 years, all improvements on Chandra et al.'s PTAS come just in the constant in the exponent [19, 8], with the best running time known being $O(n^{\lceil d/\varepsilon \rceil - d})$ [8]. Unfortunately, this barrier cannot be explained by aforementioned existing lower bounds. For example, they do not rule out an $n^{O(d+1/\varepsilon)}$ -time PTAS. This brings us to the main question of this paper:

Is there a PTAS for d -dimensional knapsack with running time $n^{o(d/\varepsilon)}$?

¹ Strictly speaking, Chandra et al.'s algorithm is for the *unbounded* version where each item can be picked multiple times. However, it can be extended to the bounded case too [44].

1.1 Our Results

Our main result is the resolution of the above question in the negative, up to a polylogarithmic factor in the exponent. Unlike previous lower bounds [35, 26] which focused on two dimensions $d = 2$, or a non-parametrized number of dimensions [11], our lower bound gives a nearly tight *trade-off* between d and $\frac{1}{\varepsilon}$, for all regimes of $\frac{d}{\varepsilon}$. In the following results, n denotes the encoding size of the instance.

► **Theorem 2.** *Assuming ETH, there are constants $A, d_0 \in \mathbb{N}$, such that for any $d \geq d_0$ and any $\varepsilon \in (0, 1)$ the following holds. There is no $(1 - \varepsilon)$ -approximation algorithm for d -dimensional knapsack that runs in time $O\left(n^{r/\log^A r}\right)$, where n is the size of the input and $r = \frac{d}{\varepsilon}$.*

For example, Theorem 2 implies that there is no algorithm which for every sufficiently large d and any $\varepsilon < 1$ returns a $(1 - \varepsilon)$ -approximate solution for a d -dimensional knapsack instance in time $n^{O(d+\frac{1}{\varepsilon})}$, or $n^{O(\frac{d^{1-\delta}}{\varepsilon})}$, or even $n^{O((d+\frac{1}{\varepsilon})^{2-\delta})}$ for any $\delta \in (0, 1)$ (by setting $d = \frac{1}{\varepsilon}$). Additionally, as the lower bound in Theorem 2 considers specific values of d and ε , the result also rules out algorithms with running time of the form $f(d, \varepsilon) \cdot n^{r/\log^A r}$ for an arbitrary function f .

1.2 Technical Overview

Existing lower bounds for d -dimensional knapsack can be crudely partitioned into three categories.

- First, lower bounds for $d = 2$ dimensions [34, 41, 35, 26]. The basic technique in this setting is to reduce Subset Sum, known to be computationally hard in various settings (e.g., [14, 1]), to 2-dimensional knapsack. In broad terms, each number in the Subset Sum instance corresponds to an item. One dimension imposes an upper bound by the target value on the original sum, while the second dimension guarantees that the original sum is higher than the target value using a clever cost construction. However, these constructions are only useful for $d = 2$ using an extremely small ε , and cannot be used to reveal the asymptotic hardness of d -dimensional knapsack for larger values of d .
- The second category of lower bounds considers the setting where d is arbitrarily large, as can be seen in [11]. The lower bound in [11] relies upon a reduction from independent set which uses a fairly high number of dimensions. In particular, the result in [11] rules out the existence of a constant approximation for multidimensional knapsack, where the number of dimensions is part of the input. However, it is unclear whether the approach in [11] can be adapted to yield lower bounds for fixed d and ε .
- The third type of lower bounds deals with the parameterized complexity of the problem. In [25] the authors show that *multidimensional subset sum* does not have a parameterized algorithm when the parameter is the number of dimensions d , even if the items' costs are encoded in unary. The result is based on a reduction from the *2-constraint satisfaction problem* (2-CSP, see definition in Section 2), which embeds each edges of the 2-CSP instance into two dimensions.

The core ideas of the reduction in [25] can be easily extended to multidimensional knapsack. Combined with recent hardness-of-approximation results for constraint satisfaction problems [4, 21, 39, 22], this extension can be used to establish lower bounds on the running time of any $(1 - \varepsilon)$ -approximation algorithm for multidimensional knapsack with fixed $\varepsilon > 0$. However, such results are limited for a fixed $\varepsilon > 0$, and will not lead to the result in Theorem 2.

We build upon the existing reduction from 2-CSP [25], and extend it to attain a lower bound which works for every sufficiently large d and every ε . The core novelty of our reduction lies in a technique which embeds *multiple* constraints of the 2-CSP instance into a *constant* number of dimensions. To do so, on certain dimensions, the weights of the items are viewed as multidigit numbers on the basis of a large number $\mathcal{Q}$, and each constraint is mapped into a specific digit.

Crucially to attain hardness of approximation results, our reduction is approximation preserving. However, the quality of the approximation preservation depends linearly on the used ratio between constraints and variables in the CSP instance. Combined with the recent hardness of approximation results for 2-CSP by Bafna Karthik and Minzer [4], this leads to a lower bound which works for every sufficiently large fixed value of d and any ε .

Organization

We start with some preliminary definitions in Section 2. Then, in Section 3 we give our reduction from 2-CSP to VK and in Section 4 we prove our main lower bound for VK. An older version of this paper, with additional results, is given in [17].

2 Preliminaries

Basic definitions

For an instance $\mathcal{I}$ of a maximization problem Π , let $\text{OPT}(\mathcal{I})$ be the optimum value of $\mathcal{I}$ and let $|\mathcal{I}|$ denote the encoding size of $\mathcal{I}$. For a set S , a function $f : S \rightarrow \mathbb{N}$, and a subset $X \subseteq S$ we use the abbreviation $f(X) = \sum_{x \in X} f(x)$. For some $\alpha \in (0, 1)$, an α -*approximate* solution for $\mathcal{I}$ is a solution for $\mathcal{I}$ of value at least $\alpha \cdot \text{OPT}(\mathcal{I})$. For all $t \in \mathbb{N}$ let $[t] = \{1, 2, \dots, t\}$. Throughout this paper, let $\log$ be the logarithm with the natural logarithm base.

d -dimensional knapsack

For short, for any $d \geq 1$ we use d -VK to denote d -dimensional (vector) knapsack (see the formal definition of d -VK in Definition 1). We also use VK to denote the collection of d -VK instances over all $d \geq 1$. For some $d \in \mathbb{N}$ and a d -VK instance $\mathcal{I} = (I, p, c, B)$, let $W(\mathcal{I}) = \max_{j \in [d]} B_j$ be the maximum budget of the instance. For any $d \in \mathbb{N}$ and $\tilde{d} \in [d]$, we may assume w.l.o.g. that every $\tilde{d}$ -dimensional knapsack instance is also a d -dimensional knapsack instance by adding extra $d - \tilde{d}$ dimensions with zero budgets and zero costs for all items.

Graph theory definitions

For every graph G let $V(G)$ and $E(G)$ denote the set of vertices and edges of G , respectively. We assume that all graphs G considered in this paper are simple directed graphs with no anti-symmetric edges. For a vertex $v \in V(G)$ we use $\text{Adj}_G(v)$ to denote the set of adjacent edges to v in G . For some $\Delta \in \mathbb{N}$, a graph G is Δ -*bounded degree* if the degree of each vertex in G is at most Δ .

3-SAT and ETH

Our lower bounds are conditioned on ETH. Before we state these, recall the standard 3-SAT problem.

► **Definition 3** (3-SAT).

- **Input:** $\phi = (V, C)$, where V is a set of variables and C is a set of disjunction clauses of three variables or negation of variables in V .
- **Assignment:** A function $s : V \rightarrow \{0, 1\}$.
- **Objective:** Find an assignment satisfying a maximum number of clauses. Let $\text{SAT}(\phi)$ be the maximum number of clauses in C satisfied by a single assignment to ϕ .

Next, we state ETH [24].

► **Conjecture 4** (Exponential-Time Hypothesis (ETH)). *There is a constant $\xi > 0$ such that there is no algorithm that given a 3-SAT formula ϕ on n variables and m clauses can decide if $\text{SAT}(\phi) = \text{True}$ in time $O(2^{\xi \cdot n})$.*

2-CSP

We prove the hardness of VK via a reduction from 2-CSP, defined formally below.

► **Definition 5** (2-CSP).

- **Input:** $\Pi = (G, \Sigma, \{\pi_e\}_{e \in E(G)})$ consisting of
 - A directed constraint graph G .
 - A finite alphabet set Σ . Without the loss of generality, we always assume that $\Sigma = \{1, \dots, n\}$ for some $n \in \mathbb{N}$.
 - For each edge $e = (u, v) \in E(G)$, a constraint $\pi_e : \Sigma \times \Sigma \rightarrow \{0, 1\}$.
- **Satisfied edges:** For $e = (u, v) \in E(G)$, we say that e is satisfied by $(\sigma_u, \sigma_v) \in \Sigma \times \Sigma$ if and only if $\pi_e(\sigma_u, \sigma_v) = 1$.
- **A solution:** A function $\varphi : V(G) \rightarrow \Sigma$.
- **Value:** The value of a solution $\varphi : V(G) \rightarrow \Sigma$ is defined as the relative number of satisfied edges:

$$\text{val}(\varphi) := \frac{|\{e = (u, v) \in E(G) \mid \pi_e(\varphi(u), \varphi(v)) = 1\}|}{|E|}$$

- **Objective:** Find a solution with maximum value.

For a 2-CSP instance Π , we define by $\text{val}(\Pi)$ the value of an optimal solution for Π . In the proof of our main result, we use a recent hardness of approximation result for 2-CSPs [4]. In the following, we state a slightly stronger version of the result from [4]. We have verified with the authors that this stronger formulation does indeed follow from their work [30].

► **Theorem 6** ([4]). *Assuming ETH, there is a constant $k_0 \in \mathbb{N}$ such that the following holds. For every constant $\xi > 0$ there exist $C > 0$ and $\Delta \in \mathbb{N}$ such that for every fixed $k \geq k_0$ there is no algorithm running in time $O(n^{k/\log^C k})$ that takes as input a 2-CSP instance Π on a constant degree Δ bipartite regular graph with k vertices over an alphabet of size n and can distinguish between the following two cases:*

- **(Completeness)** $\text{val}(\Pi) = 1$.
- **(Soundness)** $\text{val}(\Pi) < \xi$.

We remark that a recent work of [22] yields a slightly worse running time, but allowing for sub-constant soundness compared to Theorem 6.

3 A reduction from 2-CSP to VK with Varying Dimensions

In this section, we give our main reduction from 2-CSP to VK. As we will see in the next section, this implies our main result (Theorem 2). Besides the 2-CSP instance, the reduction also considers an additional integer parameter F in addition to the 2-CSP instance Π . The reduction embeds multiple edges of the 2-CSP instance into a constant number of dimensions. The parameter F determines how many edges are embedded within a constant number of dimensions. Increasing F reduces the number of dimensions of the resulting instance but also weakens the soundness guarantees of the reduction. Thus, F introduces a trade-off between the number of dimensions and the approximation ratio.

► **Lemma 7** (Dimension Embedding Reduction). *There is a reduction $2\text{-CSP} \rightarrow \text{VK}$ that, given a 2-CSP instance Π whose constraint graph $G = (V, E)$ is a Δ -bounded degree graph and $F \in [1, |E|]$, returns in time $O(|\Pi|^5)$ an instance $\mathcal{I}(\Pi, F) = (I, p, c, B)$ of d -VK such that the following holds.*

1. *The number of dimensions is $d = 4 \cdot \left\lceil \frac{|E|}{F} \right\rceil$.*
2. *$|\mathcal{I}(\Pi)| = O(|\Pi|^5)$.*
3. **(Completeness)** *If $\text{val}(\Pi) = 1$, then there is a solution for $\mathcal{I}(\Pi, F)$ with profit $8 \cdot |E|$.*
4. **(Soundness)** *For every $q \in \mathbb{N}$, if there is a solution for $\mathcal{I}(\Pi, F)$ of profit at least $8 \cdot |E| - q$, then $\text{val}(\Pi) \geq 1 - 2 \cdot \Delta \cdot F \cdot \frac{q}{|E|}$.*

Before the formal construction, we provide a high level intuitive description. For this entire section, let $\Pi = (G, \Sigma, \{\pi_e\}_{e \in E})$ be a 2-CSP instance, where G is of bounded degree Δ for some fixed constant Δ , and let $F \in [1, |E|]$. In addition, let $n \in \mathbb{N}$ such that $\Sigma = \{1, \dots, n\}$. In the following, we describe a reduction that creates the *reduced* VK instance $\mathcal{I}(\Pi, F) = (I, p, c, B)$.

The construction of the reduced instance can be intuitively summarized as follows. The instance has two types of items. The first type contains all pairs of a vertex in G and an assignment for the vertex, and the second type contains all triplets of an edge and a feasible assignment for both of its endpoints. The profits, costs, and budgets are defined so that a feasible solution $S \subseteq I$ for this VK instance with a sufficiently high profit satisfy the following: for most vertices $v \in V$ and for all of their adjacent edges $e = (u, v) \in E$ it holds that (i) exactly one item (v, σ_v) is selected to S corresponding to an assignment of σ_v to the vertex v , and (ii) exactly one item of the form (e, σ_u, σ_v) is selected to S , preserving the assignment of σ_v to v . From here, it is easy to construct an assignment for the original 2-CSP instance with relatively high value.

The non-trivial part is to enforce the above using the definition of profits, costs, and budgets. Recall that the reduction receives the parameter F which controls the number of dimensions. We define four *sub-constraints* for each edge $e \in E$ and call D the set of sub-constraints over all edges. We define a partition $D_1, \dots, D_r$ of D into r *constraints*, with up to $4 \cdot F$ sub-constraints in each constraint (with possibly fewer sub-constraints in the last constraint). For each $\ell \in [r]$, we make an *embedding* of D_ℓ into only two dimensions, thus having overall $d = 2 \cdot r$ dimensions of the VK instance.

The construction of the items' costs consists of two levels. The first level associates a weight with each item, where the weight has a dimension for every sub-constraint. The weight function ensures that if a set of items has a weight of exactly $m = 3 \cdot n$ in every dimension, then the set of items represents a solution for the 2-CSP instance of value 1. On its own, this level is similar to the reduction in [25]. The second level generates the

actual costs. Here, each constraint D_ℓ (a subset of sub-constraints) is associated with two dimensions. The first dimension encodes the weights of the sub-constraints as a $|D_\ell|$ digit number on the basis of a large number $\mathcal{Q}$, where each sub-constraint in D_ℓ is assigned to a digit. The second dimension is used to enforce the property that if a solution contains multiple items associated with the sub-constraints in D_ℓ , then the weight associated with each sub-constraint must be exactly $m = 3 \cdot n$.

The Reduction from 2-CSP to VK

We proceed with the formal description of $\mathcal{I}(\Pi, F)$. As the reduction is relatively long, we describe each parameter (the set of items, profits, costs, and budgets) in separate paragraphs.

The Items

The set of items contains all pairs of a vertex in G and a possible assignment to the vertex, as well as all edges in G and a satisfying assignment to the endpoints of the edges. Formally, let $I = I_V \cup I_E$ be the set of items, where

$$I_V = V \times \Sigma$$

and

$$I_E = \{(e, \sigma_1, \sigma_2) \in E \times \Sigma \times \Sigma \mid \pi_e(\sigma_1, \sigma_2) = 1\}.$$

For any $i = (v, \sigma) \in I_V$ and $i' = (e, \sigma_1, \sigma_2) \in I_E$, with a slight abuse of notation, we use the unified notation for the first entry of the item, corresponding for a vertex or an edge, respectively, by

$$\mu(i) = v, \mu(i') = e.$$

Before describing the remaining components of the reduction, we use the following definition of *constraints*.

Constraints

Our reduction associates four sub-constraints with every edge $e \in E$ of the 2-CSP instance, and bundles $4 \cdot F$ sub-constraints into a single constraint. More formally, a *sub-constraint* is a tuple $\chi = (e, j, s) \in E \times \{1, 2\} \times \{1, 2\}$, where $e = (u_1, u_2)$ is the edge of the sub-constraint, j selects u_j as the endpoint of e associated with the sub-constraint χ , and s can be viewed as a sign selector. In our reduction, a *constraint* is a subset of sub-constraints.

Let $E_1, \dots, E_r$ be an arbitrary partition of E such that $|E_\ell| = F$ for all $\ell \in [r-1]$ and $|E_r| \leq F$; thus, $r = \lceil \frac{|E|}{F} \rceil$ and $|E_\ell| \leq F$ for all $\ell \in [r]$. Define $D_\ell = E_\ell \times \{1, 2\} \times \{1, 2\}$ as the ℓ -th constraint. We also define $D = E \times \{1, 2\} \times \{1, 2\}$ as the set of all sub-constraints. In particular, $D_1, \dots, D_r$ is a partition of D .

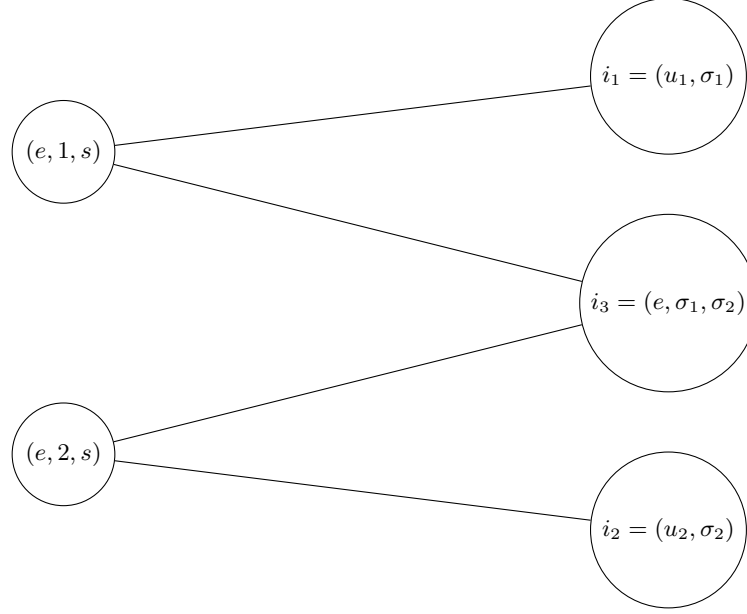
We define the following auxiliary parameter – the number of sub-constraints in which a vertex or an edge *participates* within a constraint. This parameter relates to both the profits and costs of the items. For all $\ell \in [r]$ and $v \in V$ define

$$\#_{\text{sub}}(\ell, v) = 2 \cdot |\text{Adj}_G(v) \cap E_\ell|$$

and for every $e \in E$ define

$$\#_{\text{sub}}(\ell, e) = \begin{cases} 4, & \text{if } e \in E_\ell, \\ 0, & \text{else.} \end{cases}$$

Intuitively, the values $\#sub(\ell, v)$ and $\#sub(\ell, e)$ can be viewed as the number of sub-constraints in D_ℓ in which v and e participate. An illustration of the above is given in Figure 1.



■ **Figure 1** An illustration of two sub-constraints $(e, 1, s), (e, 2, s)$ (on the left), and three items i_1, i_2, i_3 on the right. The edges in the figure between a sub-constraint and an item $i \in \{i_1, i_2, i_3\}$ indicate that $\mu(i)$ participates in the sub-constraint.

Profits

Define the profits $p : I \rightarrow \mathbb{N}$ by $p((v, \sigma)) = 2 \cdot \deg(v)$ for every $(v, \sigma) \in I_V$ and $p((e, j, s)) = 4$ for every $(e, j, s) \in I_E$.

► **Lemma 8.** For every $S \subseteq I$ and $\ell \in [r]$, we have $p(S) = \sum_{\ell \in [r]} \sum_{i \in S} \#sub(\ell, \mu(i))$.

Proof. It holds that

$$\begin{aligned}
 p(S) &= \sum_{i \in S} p(i) \\
 &= \sum_{(v, \sigma) \in S \cap I_V} 2 \cdot \deg(v) + \sum_{(e, j, s) \in S \cap I_E} 4 \\
 &= \sum_{\ell \in [r]} \sum_{(v, \sigma) \in S \cap I_V} 2 \cdot |\text{Adj}_G(v) \cap E_\ell| + \sum_{\ell \in [r]} \sum_{(e, j, s) \in S \cap I_E \text{ s.t. } e \in E_\ell} 4 \\
 &= \sum_{\ell \in [r]} \sum_{(v, \sigma) \in S \cap I_V} \#sub(\ell, v) + \sum_{\ell \in [r]} \sum_{(e, j, s) \in S \cap I_E \text{ s.t. } e \in E_\ell} \#sub(\ell, e) \\
 &= \sum_{\ell \in [r]} \sum_{i \in S} \#sub(\ell, \mu(i))
 \end{aligned}$$

The second equality holds since for every $e \in E$ there is exactly one $\ell \in [r]$ such that $e \in E_\ell$. ◀

Next, we define the costs and budgets of the reduced VK instance. As a building block, we use the following definition of a $|D|$ -dimensional *weight* function. Informally, each dimension of the weight function corresponds to a sub-constraint in D , and can be interpreted as the cost of an item on the specific sub-constraint.

Weights

Let $m = 3 \cdot n$. Define the *weight function* $w : I \rightarrow \mathbb{N}^D$ such that for every $\chi = (e', j, s) \in E \times \{1, 2\} \times \{1, 2\} = D$, where $e' = (u_1, u_2)$, define the χ -dimension w_χ as follows.

■ For every $(v, \sigma) \in I_V$ define

$$w_\chi((v, \sigma)) = \begin{cases} 0, & \text{if } v \neq u_j, \\ \sigma, & \text{if } v = u_j \text{ and } s = 1 \\ m - \sigma & \text{if } v = u_j \text{ and } s = 2. \end{cases} \quad (1)$$

■ For every $i = (e, \sigma_1, \sigma_2) \in I_E$ define

$$w_\chi(i) = \begin{cases} 0, & \text{if } e \neq e', \\ \sigma_j, & \text{if } e = e' \text{ and } s = 2 \\ m - \sigma_j & \text{if } e = e' \text{ and } s = 1. \end{cases} \quad (2)$$

Notice that the weight of an item $(v, \sigma) \in I_V$ in w_χ equals to σ if v is the endpoint of e associated with χ and $s = 1$; on the other hand, for $s = 2$ the weight the item receives is the *complement*: $m - \sigma$, considering m as a large number, strictly larger than any $\sigma' \in \Sigma$ (recall that $\Sigma = \{1, \dots, n\}$ and $m = 3 \cdot n$). The weight of an item $(e, \sigma_1, \sigma_2) \in I_E$ is defined reciprocally to the above, using σ_j if $s = 2$, and the complement $m - \sigma_j$ if $s = 1$. From here, we get the following result.

► **Lemma 9.** *Let $\varphi : V \rightarrow \Sigma$ such that $\text{val}(\varphi) = 1$, and let*

$$S = \{(v, \varphi(v)) \mid v \in V\} \cup \{(e, \varphi(u), \varphi(v)) \mid e = (u, v) \in E\}.$$

Then, for every $\chi \in D$ it holds that $w_\chi(S) = m$.

Proof. Note that $S \subseteq I$ since for every $e = (u, v) \in E$ it holds that $\pi_e(\varphi(u), \varphi(v)) = 1$ as $\text{val}(\varphi) = 1$. For any $\chi = ((u_1, u_2), j, s) \in D$, we have

$$w_\chi(S) = w_\chi(u_j, \varphi(u_j)) + w_\chi((u_1, u_2), \varphi(u_1), \varphi(u_2)) = \begin{cases} \varphi(u_j) + m - \varphi(u_j) & s = 1 \\ m - \varphi(u_j) + \varphi(u_j) & s = 2 \end{cases} = m.$$

Thus, it holds that $w_\chi(S) = m$ for every $\chi \in D$. ◀

Additional Parameters for the Costs and Budgets

Define a sufficiently large parameter as

$$Q = 3 \cdot F^2 \cdot m \cdot (|V| + |E|) \cdot |\Sigma|^2.$$

Recall that $|\Pi|$ is the encoding size of the 2-CSP instance Π . In addition, define $M = Q^{4 \cdot F}$, where the reason for the selection of these parameters becomes clear in the proofs later on. The following observation holds since $m, F, |V|, |E|, |\Sigma|$ are bounded by $|\Pi|$.

► **Observation 10.** $Q \leq 6 \cdot |\Pi|^6$ and $M \leq (6 \cdot |\Pi|)^{24 \cdot F}$.

We proceed to define the costs and budgets of the reduced VK instance.

$c_{(\ell,1)}(i)$	$w_{\chi_1}(i) = m - \sigma$	$w_{\chi_2}(i) = 0$	$w_{\chi_3}(i) = \sigma$	$w_{\chi_4}(i) = 0$
	$\text{ord}_\ell(\chi_1) = 1$	$\text{ord}_\ell(\chi_2) = 2$	$\text{ord}_\ell(\chi_3) = 3$	$\text{ord}_\ell(\chi_4) = 4$

■ **Figure 2** A partial illustration of the cost function in the reduction. The figure shows the cost of item $i = (v, \sigma) \in I_V$ in dimension $(\ell, 1)$ for some $\ell \in [r]$. The sub-constraints in the constraint D_ℓ are $D_\ell = \{\chi_1, \chi_2, \chi_3, \chi_4\}$ where $\chi_1 = (e, j = 2, s = 2)$, $j_3 = (e, j = 2, s = 1)$, $e = (u, v)$, and χ_2, χ_4 are sub-constraints corresponding to edges not adjacent to v . Thus, $\#\text{sub}(\ell, v) = 2$. The sub-constraints are ordered by $\chi_1, \chi_2, \chi_3, \chi_4$ so that $\text{ord}_\ell(\chi_1) = 1$, $\text{ord}_\ell(\chi_2) = 2$, etc. The cost of i in dimension $(\ell, 1)$ is $w_{\chi_1}(i) + w_{\chi_3}(i) \cdot \mathcal{Q}^3$. Considering this cost as a base- $\mathcal{Q}$ number, the first digit is $m - \sigma$ and the third digit is σ . This is illustrated as the gray area in the figure, depicting the 4-digit number in base- $\mathcal{Q}$.

Costs

The reduced instance will have $d = 2 \cdot r$ dimensions and without the loss of generality, the dimensions correspond to the set $[r] \times \{1, 2\}$, i.e., a pair of dimensions for each constraint D_ℓ . We define the cost function $c : I \rightarrow \mathbb{N}^{[r] \times \{1, 2\}}$ as follows. For every $\ell \in [r]$, let $\text{ord}_\ell : D_\ell \rightarrow [|D_\ell|]$ be an arbitrary bijection, that will be used to order the sub-constraints in each constraint arbitrarily into disjoint bits in the costs (and budgets). Let $i \in I$ and for $(\ell, t) \in [r] \times \{1, 2\}$ define

$$c_{(\ell,t)}(i) = \begin{cases} \sum_{\chi \in D_\ell} w_\chi(i) \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)}, & \text{if } t = 1, \\ M \cdot \#\text{sub}(\ell, \mu(i)) - \sum_{\chi \in D_\ell} w_\chi(i) \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)}, & \text{if } t = 2. \end{cases}$$

Observe that $\#\text{sub}(\ell, \mu(i)) = \{\chi \in D_\ell \mid w_\chi(i) > 0\}$, which implies that the costs in dimensions $(\ell, 2)$ are non-negative. For each $\ell \in [r]$, and $t = 1$ the cost $c_{(\ell,t)}(i)$ encodes the weights $w_\chi(i)$ of all the sub-constraints $\chi \in D_\ell$. The encoding simply views the cost as an $|D_\ell|$ -digit number on the basis of $\mathcal{Q}$, where the sub-constraint $\chi \in D_\ell$ is assigned to the $\text{ord}_\ell(\chi)$ -digit. The costs $c_{(\ell,t)}(i)$ for $t = 2$ are used as a complement to the costs in the $t = 1$ dimension. The two dimensions $(\ell, 1)$ and $(\ell, 2)$ are later used together to ensure properties of high profit solutions. We give an illustration of the costs in Figure 2.

The next observation will be helpful throughout this section.

► **Observation 11.** For any $S \subseteq I$ and $\ell \in [r]$, we have

- (i) $c_{\ell,1}(S) = \sum_{\chi \in D_\ell} w_\chi(S) \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)}$.
- (ii) $c_{\ell,2}(S) = M \cdot \sum_{i \in S} \#\text{sub}(\ell, \mu(i)) - \sum_{\chi \in D_\ell} w_\chi(S) \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)}$.

Budget

We define the d -dimensional budget $B \in \mathbb{N}^{[r] \times \{1, 2\}}$ for every dimension $(\ell, t) \in [r] \times \{1, 2\}$ by

$$B_{\ell,t} = \begin{cases} \sum_{\chi \in D_\ell} m \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)}, & \text{if } t = 1, \\ 8 \cdot M \cdot |E_\ell| - \sum_{\chi \in D_\ell} m \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)} & \text{if } t = 2. \end{cases}$$

This completes the definition of the reduced instance $\mathcal{I}(\Pi, F)$. To simplify the presentation, when Π, F are clear from the context, we may discard Π, F from the notations.

Clearly, given a 2-CSP instance Π with a constraint graph G and $F \in [1, |E|]$, the instance $\mathcal{I}(\Pi, F)$ can be constructed in time $|\Pi|^{O(1)}$. Next, we prove some basic properties of the reduction.

► **Lemma 12.** *For any 2-CSP instance $\Pi = (G, \Sigma, \{\pi_e\}_{e \in E})$ and $F \in [1, |E|]$ the following holds.*

1. $d = 4 \cdot \left\lceil \frac{|E|}{F} \right\rceil$.
2. $|\mathcal{I}(\Pi, F)| \leq O(|\Pi|^5)$.

Proof. The first property clearly holds as a simple observation from the construction of the reduced instance. For the second property of the lemma, note that the largest number in the costs/profits/budgets of the instance $\mathcal{I}(\Pi, F)$, i.e., $W(\mathcal{I}(\Pi, F))$, is bounded by $8 \cdot M \cdot |E_\ell|$. In addition, from Observation 10 it holds that $M \leq (6 \cdot |\Pi|)^{24 \cdot F}$. Thus,

$$W(\mathcal{I}(\Pi, F)) \leq 8 \cdot M \cdot |E_\ell| \leq 8 \cdot M \cdot |\Pi| \leq 8 \cdot (6 \cdot |\Pi|)^{24 \cdot F} \cdot |\Pi| \leq (48 \cdot |\Pi|)^{25 \cdot F}$$

Therefore, the reduced instance $\mathcal{I}(\Pi, F)$ can be encoded in space

$$O(|I| \cdot d \cdot \log W(\mathcal{I}(\Pi, F))) = O\left(\left(|V| \cdot |\Sigma| + |E| \cdot |\Sigma|^2\right) \cdot \frac{|E|}{F} \cdot F \cdot \log |\Pi|\right) \leq O(|\Pi|^5),$$

This shows the second property of the lemma. ◀

We also use the following auxiliary result.

► **Lemma 13.** *The following properties hold:*

- For every $\ell \in [r] : \sum_{a \in V \cup E} \# \text{sub}(\ell, a) = 8 \cdot |E_\ell|$
- $\sum_{\ell \in [r]} \sum_{a \in V \cup E} \# \text{sub}(\ell, a) = 8 \cdot |E|$

Proof. For a boolean valued expression A , define $\mathbb{1}_A = 1$ if A is true and $\mathbb{1}_A = 0$ if A is false. For the first property of the lemma, for all $\ell \in [r]$ it holds that

$$\begin{aligned} \sum_{a \in V \cup E} \# \text{sub}(\ell, a) &= \sum_{v \in V} \# \text{sub}(\ell, v) + \sum_{e \in E} \# \text{sub}(\ell, e) \\ &= \sum_{v \in V} 2 \cdot |\text{Adj}_G(v) \cap E_\ell| + \sum_{e \in E} 4 \cdot \mathbb{1}_{e \in E_\ell} \\ &= \sum_{e \in E} 8 \cdot \mathbb{1}_{e \in E_\ell} \\ &= 8 \cdot |E_\ell|. \end{aligned}$$

Since $E_1, \dots, E_r$ is a partition of E , the second property follows from the first property. ◀

We give below the completeness and soundness of the reduction. We start with the former, which is (relatively) straightforward.

► **Lemma 14 (Completeness).** *For any 2-CSP instance $\Pi = (G, \Sigma, \{\pi_e\}_{e \in E})$ and $F \in [1, |E|]$, if there is a solution $\varphi : V \rightarrow \Sigma$ for Π such that $\text{val}(\varphi) = 1$, then there is a solution for $\mathcal{I}(\Pi, F)$ of profit $8 \cdot |E|$.*

Proof. Let $\varphi : V \rightarrow \Sigma$ be a solution for Π satisfying that for every $e = (u, v) \in E$ it holds that $\pi_e(\varphi(u), \varphi(v)) = 1$. We use φ to define a solution for the VK instance $\mathcal{I}(\Pi, F)$ as follows. Define

$$S = \{(v, \varphi(v)) \mid v \in V\} \cup \{(e, \varphi(u), \varphi(v)) \mid e = (u, v) \in E\}.$$

50:12 On the PTAS Complexity of Multidimensional Knapsack

Note that for every $v \in V$ it holds that $(v, \varphi(v)) \in I_V$. In addition, for every $e = (u, v) \in E$ it holds that $\pi_e(\varphi(u), \varphi(v)) = 1$ by the definition of φ ; thus, $(e, \varphi(u), \varphi(v)) \in I_E$. We conclude that $S \subseteq I$ and is therefore well-defined. From Lemma 8 and the second item of Lemma 13, the profit is equal to

$$p(S) = \sum_{\ell \in [r]} \sum_{a \in V \cup E} \# \text{sub}(\ell, a) = 8 \cdot |E|.$$

To conclude, it remains to prove that S is a solution for $\mathcal{I}(\Pi, F)$; i.e., that all budget constraints are satisfied. To see this, first notice that for any $\chi = (u_1, u_2, j, s) \in D$, by Lemma 9 we have

$$w_\chi(S) = m \quad \forall \chi \in D. \quad (3)$$

Let $(\ell, t) \in [r] \times \{1, 2\}$ be a budget constraint. Consider the two cases depending on t :

■ $t = 1$. In this case, (3) and the first item of Observation 11 yield

$$c_{(\ell, t)}(S) = \sum_{\chi \in D_\ell} w_\chi(S) \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)} = \sum_{\chi \in D_\ell} m \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)} = B_{\ell, t}.$$

■ $t = 2$. In this case, the first item of Observation 11, we have

$$\begin{aligned} c_{(\ell, t)}(S) &= M \cdot \sum_{i \in S} \# \text{sub}(\ell, \mu(i)) - \sum_{\chi \in D_\ell} w_\chi(S) \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)} \\ &= M \cdot \sum_{i \in S} \# \text{sub}(\ell, \mu(i)) - \sum_{\chi \in D_\ell} m \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)} \\ &= M \cdot \sum_{a \in V \cup E} \# \text{sub}(\ell, a) - \sum_{\chi \in D_\ell} m \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)} \\ &= 8 \cdot M \cdot |E_\ell| - \sum_{\chi \in D_\ell} m \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)} \\ &= B_{\ell, t}. \end{aligned}$$

The second equality uses (3). The fourth equality uses the fact that for every $v \in V$ and for every $e \in E$ there is exactly one item in S in which v and e form the first coordinate. Thus, S is a feasible solution for $\mathcal{I}(\Pi, F)$ and this completes the proof. ◀

For the soundness, we prove that given a solution to the reduced instance $\mathcal{I}(\Pi, F)$, we can construct a solution for Π with roughly the same value/profit. Before that, we give a couple of useful properties over the weights w of any solution of $\mathcal{I}(\Pi, F)$:

► **Lemma 15.** *Let S be any solution for $\mathcal{I}(\Pi, F)$. Then, for any $\ell \in [r]$, the following holds:*

- (i) $\sum_{i \in S} \# \text{sub}(\ell, \mu(i)) \leq 8 \cdot |E_\ell|$.
- (ii) *If $\sum_{i \in S} \# \text{sub}(\ell, \mu(i)) = 8 \cdot |E_\ell|$, then $w_\chi(S) = m$ for all $\chi \in D_\ell$.*

To prove this, we use the following fact that an integer can be uniquely represented in base- N .

► **Observation 16.** *Let $N \in \mathbb{N}$ and $x_1, \dots, x_n, A \in \{0, \dots, N-1\}$ be such that $\sum_{i \in [n]} x_i \cdot N^i = \sum_{i \in [n]} A \cdot N^i$. Then, for all $i \in [n]$ it holds that $x_i = A$.*

Proof of Lemma 15. Since S is a solution for $\mathcal{I}(\Pi, F)$,

$$M \cdot 8 \cdot |E_\ell| = B_{\ell, 1} + B_{\ell, 2} \geq c_{\ell, 1}(S) + c_{\ell, 2}(S) = M \cdot \sum_{i \in S} \# \text{sub}(\ell, \mu(i)), \quad (4)$$

where the last equality follows from the two items of Observation 11. Thus, Item (i) of the lemma holds by dividing the two sides of the inequality by $M > 0$.

To prove Item (ii) of the lemma, note that for (4) to be an equality rather than just an inequality, we must have $B_{\ell,1} = c_{\ell,1}(S)$, since $B_{\ell,1} \geq c_{\ell,1}(S)$ and $B_{\ell,2} \geq c_{\ell,2}(S)$ (recall that S is a solution). Therefore,

$$\sum_{\chi \in D_\ell} m \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)} = B_{\ell,1} = c_{\ell,1}(S) = \sum_{\chi \in D_\ell} w_\chi(S) \cdot \mathcal{Q}^{\text{ord}_\ell(\chi)}. \quad (5)$$

Observe that $w_j(S) \leq |I| \cdot m < \mathcal{Q}$ due to our choice of the parameters m and $\mathcal{Q}$. From this, (5), and Observation 16, we can conclude that $w_\chi(S) = m$ for all $\chi \in D_\ell$ as required. $\blacktriangleleft$

Using Lemma 15, we can give the second direction of the reduction.

► **Lemma 17 (Soundness).** *For any 2-CSP instance Π , $F \in [1, |E|]$, and $q \in \mathbb{N}$, if there is a solution for $\mathcal{I}(\Pi, F)$ of profit at least $8 \cdot |E| - q$, then there is a solution φ for Π such that $\text{val}(\varphi) \geq 1 - \frac{2 \cdot \Delta \cdot q \cdot F}{|E|}$.*

The soundness proof goes, in high level, as follows. Given a solution S for $\mathcal{I}(\Pi, F)$ with relatively high profit, we define the set of *tight* constraints of S as all $\ell \in [r]$ satisfying $\sum_{i \in S} \# \text{sub}(\ell, \mu(i)) = 8 \cdot |E_\ell|$. We denote by V^* the set of vertices $v \in V$ whose all of their adjacent edges belong to a set E_ℓ such that ℓ is tight. We show that every $v \in V^*$ induce the following property: There is exactly one $(v, \sigma_v) \in S$, and for every $e \in \text{Adj}(v)$ there is exactly one $(e, \sigma_1, \sigma_2) \in S$, such that $\sigma_j = \sigma_v$ where $e = (u_1, u_2)$, $j \in \{1, 2\}$, and $v = u_j$. Therefore, we can define an assignment φ by setting $\varphi(v) = \sigma_v$ for vertices $v \in V^*$, and define the assignment of other vertices $v' \in V \setminus V^*$ arbitrarily. Finally, we can easily show that every constraint $(u, v) \in V^* \times V^*$ of the CSP instance Π is satisfied by φ immediately by the definition of I_E . Since S has a relatively high profit, it follows that $|V^*|$ is large implying that most of the constraints are tight and therefore we have constructed an assignment for Π with a relatively high profit.

Proof. Let $q \in \mathbb{N}$ such that there is a solution S for $\mathcal{I}(\Pi, F)$ of profit at least $8 \cdot |E| - q$. For every $e \in E$, let $\ell(e) \in [r]$ be the unique index satisfying that e belongs to $E_{\ell(e)}$. In addition, let

$$T = \left\{ \ell \in [r] \mid \sum_{i \in S} \# \text{sub}(\ell, \mu(i)) = 8 \cdot |E_\ell| \right\}$$

be the set of *tight constraints* of S . Define

$$V^* = \left\{ v \in V \mid \forall e \in \text{Adj}_G(v) : \ell(e) \in T \right\} \quad (6)$$

as the set of vertices satisfying that all of their adjacent edges belong to tight constraints. The set V^* satisfies the following crucial property.

► **Claim 18.** For every $v \in V^*$ there is exactly one $\sigma_v \in \Sigma$ such that $(v, \sigma_v) \in S$. Moreover, for every $e = (u_1, u_2) \in \text{Adj}_G(v)$ there is exactly one $(\sigma_1, \sigma_2) \in \Sigma^2$ such that $(e, \sigma_1, \sigma_2) \in S$ and $\sigma_j = \sigma_v$, where $j \in \{1, 2\}$ satisfies $v = u_j$.

Proof. By (6), it holds that $\ell(e)$ must be *tight* for every $e \in \text{Adj}_G(v)$, that is,

$$\sum_{i \in S} \# \text{sub}(\ell(e), \mu(i)) = 8 \cdot |E_{\ell(e)}|.$$

Let $e = (u_1, u_2) \in \text{Adj}_G(v)$ and let $\chi_1 = (e, j, 1)$, $\chi_2 = (e, j, 2)$ where $j \in \{1, 2\}$ is the unique index such that $u_j = v$. Thus, as $\chi_1, \chi_2 \in D_{\ell(e)}$, by Item (ii) of Lemma 15 it holds that $w_{\chi_1}(S) = m$ and $w_{\chi_2}(S) = m$. Observe that there can be at most one $\sigma_v \in \Sigma$ such that $(v, \sigma_v) \in S$: assume towards a contradiction that there are $(v, \sigma_v), (v, \sigma'_v) \in S$ such that $\sigma_v \neq \sigma'_v$, then it follow that

$$w_{\chi_2}(S) \geq w_{\chi_2}(v, \sigma_v) + w_{\chi_2}(v, \sigma'_v) = 3 \cdot n - \sigma_v + 3 \cdot n - \sigma'_v = 4 \cdot n > 3 \cdot n = m,$$

where first equality follows from the definition of w_{χ_2} (1) and the second inequality holds as $\sigma_v, \sigma'_v \leq n$. The above inequality is a contradiction that $w_{\chi_2}(S) = m$.

Analogously, if there are $(e, \sigma_1, \sigma_2), (e, \sigma'_1, \sigma'_2) \in S$ where $(\sigma_1, \sigma_2) \neq (\sigma'_1, \sigma'_2)$ then it would follow that

$$w_{\chi_1}(S) \geq 3 \cdot n - \sigma_j + 3 \cdot n - \sigma'_j \geq 4 \cdot n > 3 \cdot n = m,$$

where the first inequality uses the definition of w_{χ_1} (2), and the second inequality follows from $\sigma_j, \sigma'_j \leq n$. The above inequality is a contradiction to $w_{\chi_1}(S) = m$. Therefore, there is at most one items of the form (e, σ_1, σ_2) in S .

On the other hand, if for all $\sigma_v \in \Sigma$ it holds that $(v, \sigma_v) \notin S$, then since there is at most one $(\sigma_1, \sigma_2) \in \Sigma^2$ such that $(e, \sigma_1, \sigma_2) \in S$ we also have

$$w_{\chi_2}(S) \leq \max_{\sigma \in \Sigma} \sigma = n < 3 \cdot n = m.$$

The inequality holds since we assume that there is no $\sigma_v \in \Sigma$ it holds that $(v, \sigma_v) \notin S$, and we have shown above that there is at most one $(\sigma_1, \sigma_2) \in \Sigma^2$ such that $(e, \sigma_1, \sigma_2) \in S$. The above contradicts the fact that $w_{\chi_2}(S) = m$. Similarly, if for all $(\sigma_1, \sigma_2) \in \Sigma^2$ it holds that $(e, \sigma_1, \sigma_2) \notin S$, then it follows that

$$w_{\chi_1}(S) \leq \max_{\sigma \in \Sigma} \sigma = n < 3 \cdot n = m$$

The above equation is a contradiction that $w_{\chi_1}(S) = m$.

Overall, we have established that there is exactly one $\sigma_v \in \Sigma$ such that $(v, \sigma_v) \in S$ and there is exactly one $(\sigma_1, \sigma_2) \in \Sigma^2$ such that $(e, \sigma_1, \sigma_2) \in S$. Therefore, it remains to prove that $\sigma_j = \sigma_v$. Observe that

$$w_{\chi_1}(S) = w_{\chi_1}((v, \sigma_v)) + w_{\chi_1}((e, \sigma_1, \sigma_2)) = \sigma_v + 3 \cdot n - \sigma_j.$$

Since $w_{\chi_1}(S) = m = 3 \cdot n$, we conclude that $\sigma_j = \sigma_v$. The statement of the claim follows. $\triangleleft$

For all $v \in V^*$ denote by $\sigma_v \in \Sigma$ the unique symbol satisfying that $(v, \sigma_v) \in S$; by Claim 18 it holds that σ_v is well defined for every $v \in V^*$. Define $\varphi : V \rightarrow \Sigma$ by $\varphi(v) = \sigma_v$ for every $v \in V^*$, and for every $v \in V \setminus V^*$ define $\varphi(v)$ arbitrarily as some symbol in Σ . We start with the following claim.

$\triangleright$ **Claim 19.** For every $e = (u, v) \in E$ such that $u, v \in V^*$ it holds that $\pi_e(\varphi(u), \varphi(v)) = 1$.

Proof. Let $e = (u, v) \in E$ such that $u, v \in V^*$. Thus, it follows that $\varphi(u) = \sigma_u$ and $\varphi(v) = \sigma_v$, where σ_u, σ_v are the unique symbols such that $(u, \sigma_u) \in S$ and $(v, \sigma_v) \in S$, respectively, by Claim 18. Moreover, as $e \in \text{Adj}_G(v)$ and $e \in \text{Adj}_G(u)$, by Claim 18 it follows that $(e, \sigma_u, \sigma_v) \in S \cap I_E$. Since $I_E = \{(e, \sigma_1, \sigma_2) \in E \times \Sigma \times \Sigma \mid \pi_e(\sigma_1, \sigma_2) = 1\}$ it immediately follows that $\pi_e(\varphi(u), \varphi(v)) = \pi_e(\sigma_u, \sigma_v) = 1$ as required. $\triangleleft$

It remains to give a lower bound on the value of φ . To do so, we need the following upper bound on the number of non-tight indices $\ell \in [r]$.

▷ Claim 20. $|[r] \setminus T| \leq q$.

Proof. First, from Lemma 8, we have

$$\begin{aligned} p(S) &= \sum_{\ell \in [r]} \sum_{i \in S} \#_{\text{sub}}(\ell, \mu(i)) \\ &= \sum_{\ell \in T} \sum_{i \in S} \#_{\text{sub}}(\ell, \mu(i)) + \sum_{\ell \in [r] \setminus T} \sum_{i \in S} \#_{\text{sub}}(\ell, \mu(i)) \\ &= \sum_{\ell \in T} 8 \cdot |E_\ell| + \sum_{\ell \in [r] \setminus T} \sum_{i \in S} \#_{\text{sub}}(\ell, \mu(i)) \end{aligned}$$

Now, for every $\ell \in [r] \setminus T$, it holds that $\sum_{i \in S} \#_{\text{sub}}(\ell, \mu(i)) \neq 8 \cdot |E_\ell|$; by Item (i) of Lemma 15 and the fact that $\#_{\text{sub}}$ assigns only integral value for each item, we have that $\sum_{i \in S} \#_{\text{sub}}(\ell, \mu(i)) \leq 8 \cdot |E_\ell| - 1$. Plugging this into the above, we get

$$p(S) \leq \sum_{\ell \in T} 8 \cdot |E_\ell| + \sum_{\ell \in [r] \setminus T} (8 \cdot |E_\ell| - 1) = \sum_{\ell \in [r]} 8 \cdot |E_\ell| - |[r] \setminus T| = 8 \cdot |E| - |[r] \setminus T|,$$

Finally, the claim follows since we assume that $p(S) \geq 8 \cdot |E| - q$. ◁

To conclude the soundness proof, observe that

$$\begin{aligned} |V \setminus V^*| &= \left| \left\{ v \in V \mid \exists e \in \text{Adj}_G(v) \text{ s.t. } \ell(e) \in [r] \setminus T \right\} \right| \\ &\leq \sum_{\ell \in [r] \setminus T} 2 \cdot |E_\ell| \\ &\leq \sum_{\ell \in [r] \setminus T} 2 \cdot F \\ &= |[r] \setminus T| \cdot 2 \cdot F \\ &\leq 2 \cdot q \cdot F. \end{aligned} \tag{7}$$

The first inequality holds as every $e \in E$ such that $\ell(e) \in [r] \setminus T$ can add up to 2 vertices (its endpoints) to $V \setminus V^*$. The second inequality holds as $|E_\ell| \leq F$ due to the construction of the partition $E_1, \dots, E_r$ of E . The last inequality uses Claim 20. Hence,

$$\begin{aligned} \text{val}(\varphi) &= \frac{|\{e = (u, v) \in E \mid \pi_e(\varphi(u), \varphi(v)) = 1\}|}{|E|} \\ &\geq \frac{|\{e = (u, v) \in E \cap (V^* \times V^*)\}|}{|E|} \\ &= 1 - \frac{|\{e = (u, v) \in E \setminus (V^* \times V^*)\}|}{|E|} \\ &\geq 1 - \frac{\Delta \cdot |V \setminus V^*|}{|E|} \\ &\geq 1 - \frac{2 \cdot \Delta \cdot q \cdot F}{|E|}. \end{aligned}$$

The first inequality follows from Claim 19. The second inequality holds since G has a maximum degree Δ ; hence, for each $v \in V \setminus V^*$ there are at most Δ adjacent edges that do not belong to $V^* \times V^*$ and we conclude that $|E \setminus (V^* \times V^*)| \leq \Delta \cdot |V \setminus V^*|$. The last inequality follows from (7). By the above inequality, the soundness proof follows. ◀

The above lemmas give the statement of the reduction.

Proof of Lemma 7. The proof follows from Lemma 12, Lemma 14, and Lemma 17. ◀

4

 Proofs of the Main Lower Bound

In this section, we prove Theorem 2 relying on the reduction presented in Section 3 and the strong lower bound for 2-CSP of [4], formalized in Theorem 6.

► **Theorem (2).** *Assuming ETH, there is a constant $d_0 \in \mathbb{N}$, such that for any $d \geq d_0$ and any $\varepsilon \in (0, 1)$ the following holds. There exists a constant $A > 0$ such that there is no $(1 - \varepsilon)$ -approximation algorithm for d -dimensional knapsack that runs in time $O\left(n^{r/\log^A r}\right)$, where n is the size of the input and $r = \frac{d}{\varepsilon}$.*

Proof of Theorem 2. The proof can be summarized in high level as follows. First, we define a set of parameters. Then, relying on these parameters, we assume towards a contradiction the existence of an approximation algorithm VK-ALG for d -VK, for a sufficiently large d (see the concrete definitions below). Then, based on the existence of VK-ALG we give a reduction called CSP-ALG from 2-CSP to d -VK. Finally, we use the reduction to reach a contradiction to the existing lower bound for 2-CSP, given in Theorem 6. For readability, the proofs of the technical claims within the proof are deferred to Section 4.1.

Assume that ETH holds. Define $\xi = \frac{1}{4}$ and let $C > 0$ and $\Delta \in \mathbb{N}$ be the constants promised by Theorem 6 for the error parameter ξ . Let $A = C + 1$. We define several constants as follows. Let k_0 be the constant promised by Theorem 6. Let $B = 10 \cdot 2048 \cdot \Delta^2$ and let k_2 be the integer promised by the following claim.

▷ **Claim 21.** There is an integer $k_2 \geq k_0$ such that for every $k' \geq k_2$ the following holds.

- $\frac{k'}{\log^C k'} \geq 10$.
- $\frac{B \cdot k'}{\log^A k'} \leq \frac{k'}{\log^C k'}$.

Let $d_0 = 4096 \cdot \Delta^2 \cdot k_2$. Assume, towards a contradiction that there are $d \geq d_0$ and $\varepsilon \in (0, 1)$ such that there is a $(1 - \varepsilon)$ -approximation algorithm VK-ALG for d -dimensional knapsack that runs in time $O\left(N^{r/\log^A r}\right)$, where N is the size of the input and $r = \frac{d}{\varepsilon}$. To reach a contradiction to Theorem 6, we define the following reduction from 2-CSP on specific parameters. Define $d' = \frac{d}{2}$, and define $k = \left\lfloor \frac{d'}{\varepsilon \cdot 1024 \cdot \Delta^2} \right\rfloor$. It holds that k is sufficiently large by the following claim.

▷ **Claim 22.** $k \geq k_2 \geq k_0$.

According to Theorem 6, in the following we will consider only 2-CSP instances with k variables and $m = \Delta \cdot k$ constraints. Define $F = \left\lceil \frac{4 \cdot m}{d'} \right\rceil$. The following claim shows that F satisfies the conditions of Lemma 7.

▷ **Claim 23.** $F \in [m]$ and $d \geq 4 \cdot \left\lceil \frac{m}{F} \right\rceil$.

Recall that by Theorem 6 there is no algorithm running in time $O\left(n^{k/\log^C k}\right)$ that takes as input a 2-CSP instance Π on a constant degree Δ bipartite regular graph with k vertices over an alphabet of size n and can distinguish between the following two cases:

- **(Completeness)** $\text{val}(\Pi) = 1$.
- **(Soundness)** $\text{val}(\Pi) < \xi$.

By our assumption, there is a $(1 - \varepsilon)$ -approximation algorithm VK-ALG for d -dimensional knapsack that runs in time $O\left(N^{r/\log^A r}\right)$, where N is the size of the input. Using VK-ALG, we define the following algorithm CSP-ALG that decides between the above two

cases on 2-CSP instances with k variables on bipartite regular graphs of degree Δ . Let $\Pi = (G, \Sigma, \{\pi_e\}_{e \in E})$, where $G = (V, E)$, be such a 2-CSP instance and define CSP-ALG on Π as follows.

1. Compute the reduced d -VK instance $\mathcal{I}(\Pi, F)$ using Lemma 7
2. Execute VK-ALG on instance $\mathcal{I}(\Pi, F)$ and let p be the profit of the resulting solution
3. If $p \geq 8 \cdot |E| \cdot (1 - \varepsilon)$: Decide $\text{val}(\Pi) = 1$ and else decide $\text{val}(\Pi) < \xi$.

By Claim 23 and Lemma 7, the number of dimensions of the instance $\mathcal{I}(\Pi, F)$ is at most d , and recall we can treat this instance as a d -VK instance by padding the extra dimensions in the cost vectors with zeros for all items. We conclude with the following two claims that the existence of the above algorithm is a contradiction to Theorem 6.

▷ **Claim 24.** For every given 2-CSP instance $\Pi = (G, \Sigma, \{\pi_e\}_{e \in E})$ on a constant degree Δ bipartite regular graph with k vertices, Algorithm CSP-ALG correctly decides if $\text{val}(\Pi) = 1$ or $\text{val}(\Pi) < \xi$.

Proof. Observe that $\mathcal{I}(\Pi, F)$ is a well defined d -VK instance by Lemma 7 and Claim 23. Consider the following cases to show that CSP-ALG correctly decides Π .

- If $\text{val}(\Pi) = 1$. Then, by Lemma 7 there is a solution for $\mathcal{I}(\Pi, F)$ with profit $8 \cdot |E|$; thus, as VK-ALG is a $(1 - \varepsilon)$ -approximation algorithm, it follows that $p \geq 8 \cdot |E| \cdot (1 - \varepsilon)$ and consequently CSP-ALG decides $\text{val}(\Pi) = 1$.
- If $\text{val}(\Pi) < \xi$. We use the following claim.

▷ **Claim 25.** $1 - 2 \cdot \Delta \cdot F \cdot \frac{8 \cdot |E| \cdot \varepsilon}{|E|} > \xi$.

Then, by Claim 25 it follows that

$$\text{val}(\Pi) < \xi < 1 - 2 \cdot \Delta \cdot F \cdot \frac{8 \cdot |E| \cdot \varepsilon}{|E|}.$$

Therefore, by Lemma 7, for $q = \varepsilon \cdot 8 \cdot |E|$, there is no solution for $\mathcal{I}(\Pi, F)$ with profit at least $8 \cdot |E| - q = 8 \cdot |E| \cdot (1 - \varepsilon)$. Hence, $p < 8 \cdot |E| \cdot (1 - \varepsilon)$ and consequently CSP-ALG decides $\text{val}(\Pi) < \xi$.

The above two cases suffice to conclude the proof of the claim. ◁

In the following, we analyze the running time of CSP-ALG.

▷ **Claim 26.** For every given 2-CSP instance $\Pi = (G, \Sigma, \{\pi_e\}_{e \in E})$ on a constant degree Δ bipartite regular graph with k vertices, Algorithm CSP-ALG decides Π in time $O\left(n^{k/\log^C k}\right)$, where n is the size of the alphabet.

Proof. We first bound the running time of Step 1 of the algorithm, that constructs $\mathcal{I}(\Pi, F)$ (note that F does not depend on Π). Let $n = |\Sigma|$. The running time of this step is $O(|\Pi|^5)$ by Lemma 7. Note that the encoding size of Π can be bounded by $|\Pi| = O(n^2)$ since for each $e \in E$ we can encode π_e using $O(n^2)$ bits, and there are $m = \Delta \cdot k = O(1)$ such functions (recall that k is fixed for CSP-ALG and is not part of the input to the algorithm). Thus, the running time of Step 1 is

$$O(|\Pi|^5) = O((n^2)^5) = O(n^{10}).$$

In addition, Step 2 can be computed in time

$$\begin{aligned} O\left(|\mathcal{I}(\Pi, F)|^{r/\log^A r}\right) &= O\left(|\Pi|^{5 \cdot r/\log^A r}\right) = O\left(n^{10 \cdot r/\log^A r}\right) = O\left(n^{10 \cdot (2048 \cdot \Delta^2 \cdot k)/\log^A r}\right) \\ &= O\left(n^{B \cdot k/\log^A k}\right) = O\left(n^{k/\log^C k}\right). \end{aligned}$$

The first equality follows from $|\mathcal{I}(\Pi, F)| = O(|\Pi|^5)$ by Lemma 7. The second equality holds since the encoding size of Π is bounded by $|\Pi| = O(n^2)$ as explained above. The third equality uses that $k = \lfloor \frac{r}{\Delta^2 \cdot 2048} \rfloor = \lfloor \frac{d'}{\varepsilon \cdot \Delta^2 \cdot 1024} \rfloor$; thus, $r \leq 2048 \cdot \Delta^2 \cdot k$. The fourth inequality follows because $r \geq k \geq 1$ and by the monotonicity of the function $\log^A x$ in the domain $[1, \infty)$ (recall that $A \geq 1$ since $C > 0$). The last equality uses Claim 21 as $k \geq k_2$.

Thus, overall the running time of CSP-ALG on instance Π can be bounded by

$$O(n^{10}) + O\left(n^{k/\log^C k}\right) = O\left(n^{k/\log^C k}\right).$$

The equality follows from the fact that $\frac{k}{\log^C k} \geq 10$ from Claim 21, using that $k \geq k_2$. $\triangleleft$

By Claim 24 and Claim 26 we reach contradiction to Theorem 6 and the proof of the theorem follows. $\blacktriangleleft$

4.1 Deferred Technical Proofs

We give below the deferred proofs of the technical claims used throughout the proof of Theorem 2.

$\triangleright$ **Claim 21.** There is an integer $k_2 \geq k_0$ such that for every $k' \geq k_2$ the following holds.

- $\frac{k'}{\log^C k'} \geq 10$.
- $\frac{B \cdot k'}{\log^A k'} \leq \frac{k'}{\log^C k'}$.

Proof. Let $k_1 \in \mathbb{N}$ be the minimum integer satisfying that for every $k' \geq k_1$ it holds that $\frac{k'}{\log^C k'} \geq 10$. Clearly, there is such an integer since C is fixed. Recall that $B = 10 \cdot 2048 \cdot \Delta^2$ and define $k_2 = \max\{k_0, k_1, e^B\}$. Since $k_2 \geq e^B$, it follows that $\log k_2 \geq B$; thus, $\log^A k_2 \geq B \cdot \log^C k_2$ and for every $k' \geq k_2$ it holds that $\frac{B \cdot k'}{\log^A k'} \leq \frac{k'}{\log^C k'}$ as required. $\triangleleft$

$\triangleright$ **Claim 22.** $k \geq k_2 \geq k_0$.

Proof. Since we have

$$\frac{d'}{\varepsilon} \geq d' = \frac{d}{2} \geq \frac{d_0}{2} = 2048 \cdot \Delta^2 \cdot k_2,$$

it follows that $\frac{d'}{\varepsilon \cdot 2048 \cdot \Delta^2} \geq k_2$; thus,

$$k = \left\lfloor \frac{d'}{\varepsilon \cdot 1024 \cdot \Delta^2} \right\rfloor \geq \frac{d'}{\varepsilon \cdot 2048 \cdot \Delta^2} \geq k_2 \geq k_0.$$

The claim follows. $\triangleleft$

$\triangleright$ **Claim 23.** $F \in [m]$ and $d \geq 4 \cdot \lceil \frac{m}{F} \rceil$.

Proof. Since $d' \geq \frac{d_0}{2} \geq 4$, it follows that $F \leq m$. For the second direction, $F = \lceil \frac{4m}{d'} \rceil \geq 1$ as we round up and $m = \Delta \cdot k \geq 1$ (using that $d' \geq \frac{d_0}{2} \geq 1024 \cdot \Delta^2$). Thus, $F \in [m]$. In addition,

$$4 \cdot \left\lceil \frac{m}{F} \right\rceil = 4 \cdot \left\lceil \frac{m}{\lceil \frac{4m}{d'} \rceil} \right\rceil \leq 4 \left\lceil \frac{m}{\frac{4m}{d'}} \right\rceil = 4 \cdot \left\lceil \frac{d'}{4} \right\rceil \leq 4 \cdot \frac{2 \cdot d'}{4} = 2 \cdot d' = d.$$

The above gives the statement of the claim. $\triangleleft$

$\triangleright$ **Claim 25.** $1 - 2 \cdot \Delta \cdot F \cdot \frac{8 \cdot |E| \cdot \varepsilon}{|E|} > \xi$.

Proof. Recall that $\frac{d'}{\varepsilon} \geq d' \geq \frac{d_0}{2} \geq 2048 \cdot \Delta^2$. Thus, $k = \left\lfloor \frac{d'}{\varepsilon \cdot 1024 \cdot \Delta^2} \right\rfloor \geq 1$ and it follows that

$$\left\lfloor \frac{d'}{\varepsilon \cdot 1024 \cdot \Delta^2} \right\rfloor \geq \frac{d'}{\varepsilon \cdot 512 \cdot \Delta^2}.$$

Thus, using the definition of $F = \lceil \frac{4 \cdot \Delta \cdot k}{d'} \rceil$:

$$\begin{aligned} 2 \cdot \Delta \cdot F \cdot \frac{8 \cdot |E| \cdot \varepsilon}{|E|} &= 16 \cdot \Delta \cdot F \cdot \varepsilon = 16 \cdot \Delta \cdot \left\lceil \frac{4 \cdot \Delta \cdot \left\lfloor \frac{d'}{\varepsilon \cdot 1024 \cdot \Delta^2} \right\rfloor}{d'} \right\rceil \cdot \varepsilon \\ &\leq 16 \cdot \Delta \cdot 4 \cdot 2 \cdot \frac{\Delta \cdot \left\lfloor \frac{d'}{\varepsilon \cdot 1024 \cdot \Delta^2} \right\rfloor}{d'} \cdot \varepsilon \leq \frac{128 \cdot \Delta^2 \cdot d' \cdot \varepsilon}{1024 \cdot \Delta^2 \cdot d' \cdot \varepsilon} \leq \frac{1}{2} \end{aligned}$$

Therefore,

$$1 - 2 \cdot \Delta \cdot F \cdot \frac{8 \cdot |E| \cdot \varepsilon}{|E|} \geq 1 - \frac{1}{2} > \xi.$$

The proof follows. $\triangleleft$

References

- 1 Amir Abboud, Karl Bringmann, Danny Hermelin, and Dvir Shabtay. Seth-based lower bounds for subset sum and bicriteria path. *ACM Transactions on Algorithms (TALG)*, 18(1):1–22, 2022. doi:10.1145/3450524.
- 2 Inès Alaya, Christine Solnon, and Khaled GHÉDIRA. Ant algorithm for the multidimensional knapsack problem. In *BIOMA*, pages 63–72, 2004.
- 3 Md. Abul Kalam Azad, Ana Maria A. C. Rocha, and Edite M. G. P. Fernandes. Improved binary artificial fish swarm algorithm for the 0-1 multidimensional knapsack problems. *Swarm Evol. Comput.*, 14:66–75, 2014. doi:10.1016/J.SWEVO.2013.09.002.
- 4 Mitali Bafna, CS Karthik, and Dor Minzer. Near optimal constant inapproximability under eth for fundamental problems in parameterized complexity. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 2118–2129, 2025.
- 5 Stefan Balev, Nicola Yanev, Arnaud Fréville, and Rumen Andonov. A dynamic programming based reduction procedure for the multidimensional 0–1 knapsack problem. *European journal of operational research*, 186(1):63–76, 2008. doi:10.1016/J.EJOR.2006.02.058.
- 6 Vincent Boyer, Moussa Elkihel, and Didier El Baz. Heuristics for the 0–1 multidimensional knapsack problem. *European Journal of Operational Research*, 199(3):658–664, 2009. doi:10.1016/J.EJOR.2007.06.068.
- 7 Karthik C. S., Daniel Marx, Marcin Pilipczuk, and Uéverton Souza. Conditional lower bounds for sparse parameterized 2-csp: A streamlined proof. *arXiv e-prints*, pages arXiv–2311, 2023.

- 8 Alberto Caprara, Hans Kellerer, Ulrich Pferschy, and David Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *Eur. J. Oper. Res.*, 123(2):333–345, 2000. doi:10.1016/S0377-2217(99)00261-1.
- 9 Timothy M. Chan. Approximation schemes for 0-1 knapsack. In *SOSA*, pages 5:1–5:12, 2018. doi:10.4230/OASIcs.SOSA.2018.5.
- 10 Ashok K. Chandra, Daniel S. Hirschberg, and C. K. Wong. Approximate algorithms for some generalized knapsack problems. *Theor. Comput. Sci.*, 3(3):293–304, 1976. doi:10.1016/0304-3975(76)90048-7.
- 11 Chandra Chekuri and Sanjeev Khanna. On multidimensional packing problems. *SIAM journal on computing*, 33(4):837–851, 2004. doi:10.1137/S0097539799356265.
- 12 Lin Chen, Jiayi Lian, Yuchen Mao, and Guochuan Zhang. A nearly quadratic-time FPTAS for knapsack. *CoRR*, abs/2308.07821, 2023. doi:10.48550/arXiv.2308.07821.
- 13 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009. URL: <http://mitpress.mit.edu/books/introduction-algorithms>.
- 14 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as cnf-sat. *ACM Transactions on Algorithms (TALG)*, 12(3):1–24, 2016. doi:10.1145/2925416.
- 15 Marek Cygan, Marcin Mucha, Karol Wegrzycki, and Michal Włodarczyk. On problems equivalent to (min, +)-convolution. *ACM Trans. Algorithms*, 15(1):14:1–14:25, 2019. doi:10.1145/3293465.
- 16 Mingyang Deng, Ce Jin, and Xiao Mao. Approximating knapsack and partition via dense subset sums. In *SODA*, pages 2961–2979, 2023. doi:10.1137/1.9781611977554.CH113.
- 17 Ilan Doron-Arad, Ariel Kulik, and Pasin Manurangsi. Fine grained lower bounds for multidimensional knapsack. *arXiv preprint arXiv:2407.10146*, 2024. doi:10.48550/arXiv.2407.10146.
- 18 Arnaud Fréville. The multidimensional 0-1 knapsack problem: An overview. *Eur. J. Oper. Res.*, 155(1):1–21, 2004. doi:10.1016/S0377-2217(03)00274-1.
- 19 A.M. Frieze and M.R.B. Clarke. Approximation algorithms for the m-dimensional 0-1 knapsack problem: Worst-case and probabilistic analyses. *European Journal of Operational Research*, 15(1):100–109, 1984. doi:10.1016/0377-2217(84)90053-5.
- 20 Kilian Grage and Klaus Jansen. Convolution and knapsack in higher dimensions. *arXiv preprint arXiv:2403.16117*, 2024. doi:10.48550/arXiv.2403.16117.
- 21 Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, and Kewen Wu. Parameterized inapproximability hypothesis under exponential time hypothesis. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 24–35, 2024. doi:10.1145/3618260.3649771.
- 22 Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, and Kewen Wu. Almost optimal time lower bound for approximating parameterized clique, csp, and more, under eth. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 2136–2144, 2025. doi:10.1145/3717823.3718130.
- 23 Oscar H. Ibarra and Chul E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22(4):463–468, 1975. doi:10.1145/321906.321909.
- 24 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/JCSS.2000.1727.
- 25 Tanmay Inamdar, Daniel Lokshtanov, Saket Saurabh, and Vaishali Surianarayanan. Parameterized Complexity of Fair Bisection: (FPT-Approximation meets Unbreakability). In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63:1–63:17, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2023.63.

- 26 Klaus Jansen, Felix Land, and Kati Land. Bounding the running time of algorithms for scheduling and packing problems. *SIAM J. Discret. Math.*, 30(1):343–366, 2016. doi:10.1137/140952636.
- 27 Ce Jin. An improved FPTAS for 0-1 knapsack. In *ICALP*, pages 76:1–76:14, 2019. doi:10.4230/LIPIcs.ICALP.2019.76.
- 28 David S. Johnson. Approximation algorithms for combinatorial problems. In *STOC*, pages 38–49, 1973. doi:10.1145/800125.804034.
- 29 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations*, pages 85–103, 1972. doi:10.1007/978-1-4684-2001-2_9.
- 30 C. S. Karthik. Personal communication, 2025.
- 31 Liangjun Ke, Zuren Feng, Zhigang Ren, and Xiaoliang Wei. An ant colony optimization approach for the multidimensional knapsack problem. *Journal of Heuristics*, 16:65–83, 2010. doi:10.1007/S10732-008-9087-X.
- 32 Hans Kellerer and Ulrich Pferschy. A new fully polynomial time approximation scheme for the knapsack problem. *J. Comb. Optim.*, 3(1):59–71, 1999. doi:10.1023/A:1009813105532.
- 33 Hans Kellerer and Ulrich Pferschy. Improved dynamic programming in connection with an FPTAS for the knapsack problem. *J. Comb. Optim.*, 8(1):5–11, 2004. doi:10.1023/B:JOCO.0000021934.29833.6B.
- 34 Bernhard Korte and Rainer Schrader. On the existence of fast approximation schemes. In *Nonlinear Programming 4*, pages 415–437. Academic Press, 1981. doi:10.1016/B978-0-12-468662-5.50020-3.
- 35 Ariel Kulik and Hadas Shachnai. There is no EPTAS for two-dimensional knapsack. *Inf. Process. Lett.*, 110(16):707–710, 2010. doi:10.1016/J.IPL.2010.05.031.
- 36 Marvin Künnemann, Ramamohan Paturi, and Stefan Schneider. On the fine-grained complexity of one-dimensional dynamic programming. In *ICALP*, pages 21:1–21:15, 2017. doi:10.4230/LIPIcs.ICALP.2017.21.
- 37 Eugene L. Lawler. Fast approximation algorithms for knapsack problems. *Math. Oper. Res.*, 4(4):339–356, 1979. doi:10.1287/MOOR.4.4.339.
- 38 Soh-Yee Lee and Yoon-Teck Bau. An ant colony optimization approach for solving the multidimensional knapsack problem. In *ICCIS*, pages 441–446, 2012. doi:10.1109/ICCISci.2012.6297286.
- 39 Daniel Lokshantov, MS Ramanujan, Saket Saurab, and Meirav Zehavi. Parameterized complexity and approximability of directed odd cycle transversal. In *SODA*, pages 2181–2200, 2020.
- 40 James H Lorie and Leonard J Savage. Three problems in rationing capital. *The journal of business*, 28(4):229–239, 1955.
- 41 Michael J. Magazine and Maw-Sheng Chern. A note on approximation schemes for multidimensional knapsack problems. *Math. Oper. Res.*, 9(2):244–247, 1984. doi:10.1287/MOOR.9.2.244.
- 42 Xiao Mao. $(1-\epsilon)$ -approximation of knapsack in nearly quadratic time. *CoRR*, abs/2308.07004, 2023. doi:10.48550/arXiv.2308.07004.
- 43 Harry M Markowitz and Alan S Manne. On the solution of discrete programming problems. *Econometrica: journal of the Econometric Society*, pages 84–110, 1957.
- 44 Osman Oguz and MJ Magazine. A polynomial time approximation algorithm for the multidimensional 0/1 knapsack problem. *Univ. Waterloo Working Paper*, 1980.
- 45 Abdellah Rezoug, Mohamed Bader-El-Den, and Dalila Boughaci. *Hybrid Genetic Algorithms to Solve the Multidimensional Knapsack Problem*, pages 235–250. Springer International Publishing, Cham, 2019. doi:10.1007/978-3-319-95104-1_15.
- 46 Donguk Rhee. *Faster fully polynomial approximation schemes for knapsack problems*. PhD thesis, Massachusetts Institute of Technology, 2015.

- 47 Sara Sabba and Salim Chikhi. A discrete binary version of bat algorithm for multidimensional knapsack problem. *Int. J. Bio Inspired Comput.*, 6(2):140–152, 2014. doi:10.1504/IJBIC.2014.060598.
- 48 Sartaj Sahni. Approximate algorithms for the 0/1 knapsack problem. *J. ACM*, 22(1):115–124, 1975. doi:10.1145/321864.321873.
- 49 H Martin Weingartner and David N Ness. Methods for the solution of the multidimensional 0/1 knapsack problem. *Operations Research*, 15(1):83–103, 1967. doi:10.1287/OPRE.15.1.83.
- 50 David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. URL: http://www.cambridge.org/de/knowledge/isbn/item5759340/?site_locale=de_DE.
- 51 Hong-Fang Yan, Ci-Yun Cai, De-Huai Liu, and Min-Xia Zhang. Water wave optimization for the multidimensional knapsack problem. In *ICIC*, volume 11644, pages 688–699, 2019. doi:10.1007/978-3-030-26969-2_65.