

Approximation Schemes for Deal Splitting and Covering Integer Programs with Multiplicity Constraints*

Ariel Kulik[†] Hadas Shachnai[‡] Oded Shmueli[§] Robert Sayegh[¶]

Abstract

We consider the problem of splitting an order for R goods, $R \geq 1$, among a set of sellers, each having *bounded* amounts of the goods, so as to minimize the total cost of the deal. In *deal splitting with packages (DSP)*, the sellers offer packages containing combinations of the goods; in *deal splitting with price tables (DST)*, the buyer can generate such combinations using price tables. Our problems, which often occur in online reverse auctions, generalize *covering integer programs with multiplicity constraints (CIP)*, where we must fill up an R -dimensional bin by selecting (with bounded number of repetitions) from a set of R -dimensional items, such that the overall cost is minimized. Thus, both DSP and DST are NP-hard, already for a single good, and hard to approximate for arbitrary number of goods.

In this paper we focus on finding efficient approximations for DSP and DST instances where the number of goods is some fixed constant. In particular, we develop *polynomial time approximation schemes (PTAS)* for several subclasses of instances of practical interest. Our results include a PTAS for CIP in fixed dimension, and a more efficient (combinatorial) scheme for CIP_∞ , where the multiplicity constraints are omitted. Our approximation scheme for CIP_∞ is based on a non-trivial application of the fast scheme for the *fractional covering problem*, proposed by Fleischer (*Proc. ACM-SIAM Symposium on Discrete Algorithm, 2004*).

Keywords: Deal splitting, reverse auctions, covering integer programs, multi-dimensional knapsack, approximation algorithms.

1 Introduction

An increasing number of companies are using online *reverse auctions* for their sourcing activities (see, e.g., in [2], and a detailed survey in [26]). In reverse auctions, multiple sellers bid for a

*An extended abstract of this paper appeared in proceedings of the 2nd Workshop on Approximation and Online Algorithms (WAOA'04).

[†]Computer Science Dept., Technion, Haifa 32000, Israel. E-mail: kulik@cs.technion.ac.il.

[‡]Computer Science Dept., Technion, Haifa 32000, Israel. E-mail: hadas@cs.technion.ac.il. Part of this work was done while the author was on leave in Bell Laboratories, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974.

[§]Computer Science Dept., Technion, Haifa 32000, Israel. E-mail: oshmu@cs.technion.ac.il.

[¶]22 Anilevitch St. Haifa 35025, Israel. E-mail: abufayez@hotmail.com.

contract from a buyer for selling goods and/or services. We consider the *deal splitting* problems arising in these reverse auctions. Suppose that a buyer needs to order multiple units from a set of R goods. The number of units required from the j -th good, $1 \leq j \leq R$, is $n_j \geq 1$. The goods can be obtained from m sellers, S_1, \dots, S_m . Each seller offers a certain amount from each good (or some combination of the goods); the maximum number of units of the j -th good available from S_i is T_{ji} , $1 \leq j \leq R$, $1 \leq i \leq m$. A deal may split the order for the goods among a subset of the sellers. We say that a deal is *feasible* if (i) the number of units obtained from the j -th good is at least n_j , $1 \leq j \leq R$, and (ii) the amount of the j -th good obtained from S_i does not exceed T_{ji} , its supply from that good, $1 \leq j \leq R$, $1 \leq i \leq m$. The goal is to find a feasible deal of minimum total cost. Deal splitting naturally models a procurement auction to obtain raw materials with flexible sized lots, and many other services. We consider two variants of the problem.

In *deal splitting with packages (DSP)*, each of the sellers, S_i , offers a set of N_i packages. The ℓ -th package, p_ℓ^i , $1 \leq \ell \leq N_i$, has a non-negative cost $c(p_\ell^i)$ and is given by the R -tuple $(n_{\ell 1}^i, \dots, n_{\ell R}^i)$; that is, S_i offers in this package $0 \leq n_{\ell j}^i \leq n_j$ units from the j -th good, $1 \leq j \leq R$. We need to find a feasible deal that minimizes the total cost.

In *deal splitting with price tables (DST)*, each seller S_i , has m_i price ranges. The minimal and maximal numbers of units of the j -th good available from S_i in the ℓ -th price range are $r_{\ell j}$ and $u_{\ell j}$, respectively. The unit cost for the j -th good in the ℓ -th range is $c_{\ell j}$, $1 \leq \ell \leq m_i$, $1 \leq j \leq R$. Thus, the ℓ -th entry in the price table of S_i is given by the vector $\{(r_{\ell 1}, u_{\ell 1}, c_{\ell 1}), \dots, (r_{\ell R}, u_{\ell R}, c_{\ell R})\}$. We need to find a feasible deal in which the sale of S_i , $1 \leq i \leq m$, corresponds to a valid entry in its price table, and the total cost is minimized.

Example 1.1 Suppose that $R = 3$ and the goods are printers, cartridges and paper boxes. Table 1 gives the possible combinations of goods for the seller S_1 , specified by amounts and unit costs, in 3 price ranges (i.e., $m_1 = 3$).

Price range	Printers	Cartridges	Paper
1	(0, 2, 300)	(0, 6, 30)	(0, 9, 15)
2	(3, 5, 280)	(7, 9, 25)	(10, 100, 10)
3	(6, 20, 250)	(10, 50, 23)	(10, 100, 10)

Table 1: A price table for multiple (3) goods

Thus, if we buy 2 printers or less, the unit cost is 300, whereas the unit cost for buying $3 \leq p \leq 5$ printers is 280. A valid sale for S_1 is the combination (1, 0, 7), in which it supplies a printer and 7 paper boxes. The cost of this sale, which corresponds to the first price range, is 405.

We note that DSP is NP-hard already for $R = 1$ (by reduction from Partition) and hard to approximate within factor $\Omega(\log R)$ for arbitrary $R > 1$, as it includes as a special case the *multi-set multi-cover* problem (see in Section 1.1). For DST, we note that each price range of a seller “encodes” a possibly large number of packages (each formed by choosing the number of units from each good), as well as a simple rule for computing the price of a particular package (via the unit costs). Thus, in the special case where the price table of each seller consists of a *single* price range, which allows to form a *single* combination of the goods, we get an instance of the *constrained multi-set multi-cover* (see in Section 1.1). It follows that DST is also hard to approximate within factor $\Omega(\log R)$.

Note that DSP generalizes also *covering integer program with multiplicity constraints (CIP)*. In this core problem, we must fill up an R -dimensional bin by selecting (with bounded number of repetitions) from a set of R -dimensional items, such that the overall cost is minimized. Formally, let $A = \{a_{ji}\}$ denote the sizes of the items in the R dimensions, $1 \leq j \leq R$, $1 \leq i \leq n$; the cost of item i is $c_i \geq 0$. Let $x_i \geq 0$ denote the number of copies selected from item i , $1 \leq i \leq n$. We seek an n -vector \mathbf{x} of non-negative integers, which minimizes $c^T \mathbf{x}$, subject to the R constraints given by $A\mathbf{x} \geq b$, where $b_j \geq 0$ is the size of the bin in dimension j . In addition, we have multiplicity constraints for the vector \mathbf{x} , given by $\mathbf{x} \leq \mathbf{d}$, where $\mathbf{d} \in \{1, 2, \dots\}^n$. Recall that, in DSP, each seller S_i has T_{ji} units from the j -th good. Consider, for example, the case where $R = 2$, and suppose that S_i has $T_{1i} = 10$ units from the first good and $T_{2i} = 20$ units from the second good. S_i offers two possible packages: $p_1^i = (5, 7)$ and $p_2^i = (6, 2)$; then if we obtain two copies of p_1^i , no copies of p_2^i are available. This dependence among the packages makes DSP a generalization of CIP.¹ Indeed, an instance of CIP can be formulated as a special case of DSP, where the number of sellers, m , is equal to the number of variables in the integer program, each seller S_i offers a *single* package, and $T_{ji} = d_i a_{ji}$, $1 \leq i \leq m$, $1 \leq j \leq R$.

1.1 Relation of Deal Splitting to Set Cover

Recall that in *multi-set multi-cover* we have a set E of elements and a collection of multi-subsets of the elements; a multi-set S may contain multiple copies of some elements. Each element j , $1 \leq j \leq |E|$, has a coverage requirement, $r_j \geq 1$; each multi-set S is associated with some positive cost c_S . We need to find a collection of multi-sets that satisfy the coverage requirements, such that the total cost is minimized. Indeed, an instance of multi-set multi-cover can be formulated as the following special case of DSP. There are m sellers: S_i offers a single package p^i , such that the number of units of good j in any package p^i is at most r_j ; T_{ji} is unbounded for all i, j . We note that, in a general DSP instance, the supply of the j -th good is bounded for $1 \leq i \leq m$ and $1 \leq j \leq R$, and each seller S_i may offer a set of packages. We need to find a minimum cost deal for $R = |E|$ goods, by ordering at least $n_j = r_j$ units from the j -th good. As multi-set multi-cover is hard to approximate within factor $\Omega(\log |E|)$ [10] we get that, for arbitrary R , DSP is hard to approximate within factor $\Omega(\log R)$. When each seller offers a single package, in which we can buy T_{ji} units from the j -th good (i.e., the total supply from that good), we get the *constrained multi-set multi-cover* problem which generalizes *set cover* [34]; thus, for arbitrary R , such instances of DSP are also hard to approximate within factor $\Omega(\log R)$.

1.2 Our Results

Since our deal splitting problems are harder than set cover, the best approximation ratio that we can expect for arbitrary R is $O(\log R)$ (see, e.g., in [34]); thus, we focus on finding efficient approximations for subclasses of instances in which R is a fixed constant. We summarize below our main results.

¹In the corresponding integer program, we get dependencies among the variables that give the number of copies obtained from each package.

Deal Splitting with Packages: In Section 2.1 we develop a PTAS for instances where the i -th seller offers a set of $N_i \geq 1$ packages, $p_1^i, \dots, p_{N_i}^i$, and the buyer can obtain at most r_ℓ^i copies from p_ℓ^i , for some $r_\ell^i \geq 1$, $1 \leq \ell \leq N_i$; the total amount of the j -th good available from S_i is $T_{ji} = \sum_{\ell=1}^{N_i} n_{\ell j}^i r_\ell^i$, $1 \leq j \leq R$, $1 \leq i \leq m$. We call this version *DSP with bounded multiplicity*. As shown above, any instance of CIP can be formulated as an instance of DSP. Thus, we get a PTAS for CIP in fixed dimension.

In Section 2.1.3 we show that, in fact, a PTAS is the best we can expect for CIP in fixed dimension, and thus also for DSP with bounded multiplicity. Recall that, given an input I and a parameter $\varepsilon > 0$, an *efficient polynomial time approximation scheme* (EPTAS) is a scheme whose running time is $f(\varepsilon) \cdot \text{poly}(|\mathcal{I}|)$, where f is an arbitrary function and $|\mathcal{I}|$ is the input size, while a *fully polynomial time approximation scheme* (FPTAS) has running time that is polynomial in the input size and in $1/\varepsilon$. We show that there is no FPTAS for CIP unless $P = NP$, and that there is no EPTAS for CIP unless $W[1] = FPT$.² Both of these hardness results hold already for the case where $R = 2$.

In Section 2.2 we consider DSP instances with unbounded supply. Such instances model deals in which the buyer's need is much smaller than the supply from each of the goods. For this special case, that we call *unbounded DSP*, we develop a faster (combinatorial) scheme. This gives a combinatorial approximation scheme for CIP_∞ .

Deal Splitting with Price Tables: We develop (in Section 3) a PTAS for DST instances in which the price tables satisfy some natural properties such as *volume discount*, that is widely used in reverse auctions (see, e.g., in [21], [1]).³

Techniques: Our PTAS for unbounded DSP (in Section 2.2) is based on a non-trivial application of an FPTAS for the *fractional covering problem*, due to Fleischer [11]. We use this combinatorial scheme to obtain an *approximate* fractional solution for a *linear programming (LP)* formulation of our problem, building on a technique of Chandra et al. [3]. We show that by rounding an approximate solution for the LP we increase the cost of the optimal (integral) solution for the DSP instance only by factor of ε . Thus, we get a fast combinatorial implementation for our LP-based scheme. The overall running time of the scheme is $O(N^{\lceil 2R/\varepsilon \rceil} \cdot \frac{1}{\varepsilon^2} \log C)$, where $N = \sum_{i=1}^m N_i$ is the total number of distinct packages offered by the sellers, and $C = \max_{1 \leq i \leq N} c_i$ is the maximal cost of any package. Since unbounded DSP is equivalent to CIP_∞ , this yields a *combinatorial* approximation scheme for CIP_∞ in fixed dimension. With slight modification, we get the first combinatorial scheme for *multi-dimensional multiple choice knapsack*.

In our PTAS for DST (in Section 3), we combine the guessing technique of Chekuri and Khanna [4] with a novel application of a technique of Frieze and Clarke [9], to the minimum binary multiple choice knapsack problem in fixed dimension. Indeed, due to the constraints imposed on the solution for DST — the amounts chosen from the goods for each seller must correspond to a valid entry in its price table — we cannot apply the rounding technique of [9] to

²We refer the reader to [8] for further details on parameterized complexity, EPTAS and the complexity classes $W[1]$ and FPT .

³We elaborate on these properties in Section 3.

the fractional solution obtained by our scheme; instead, we apply non-standard rounding, which relies heavily on the mathematical properties of the price tables.

1.3 Related Work

Procurement auctions: Our deal splitting problems belong to the class of *winner determination* problems in reverse auctions. Generally, in reverse auction we have a single buyer that needs to obtain multiple goods, and a set of sellers offers *bids* for selling the goods. Bidding may follow various mechanisms (see, e.g., in [18, 24, 15]; a survey of common mechanisms is given in [35]). The DST problem with single good (i.e., $R = 1$) and price tables that satisfy the volume discount property⁴ was studied in [21]. The paper shows that DST is NP-hard already in this case and presents an FPTAS for the problem. A survey of other results for single good multi-unit auctions appears e.g., in [36]. There has been some previous work on deal splitting with *multiple* goods. The paper [29] considers DSP with bounded multiplicity, where $r_\ell^i = 1$ for $1 \leq i \leq m$, $1 \leq \ell \leq N_i$, i.e., the buyer can obtain a *single* copy of any package. The paper gives a proof of hardness for the problem, as well as an $O(\log R)$ approximation algorithm. Goossens et al. [13] considered a class of instances of DST, which satisfies a *total quantity discount* structure. While in general, the paper shows that a constant ratio approximation algorithm is unlikely to exist for such instances, for certain subclasses the authors give exact polynomial time algorithms. Other papers present either experimental studies or software that implements a given mechanism (see, e.g., [1, 2, 25, 24]). Heuristic methods and preliminary analytic results related to deal splitting are given in [33].

Multiple Choice Knapsack (MCK): As shown in Section 2.1, DSP with bounded multiplicity can be reduced to the *minimum R-dimensional binary MCK (R-MMCK)* problem. The *maximum* variant of this problem was studied since the mid-1970's (see, e.g., [23], [17], [16]). For single dimension, the best known result is a PTAS by Chandra et al. [3]. Most of the published work on the maximum multi-dimensional binary MCK presented heuristic solutions (see, e.g., [14, 12] and the references therein). Shachnai and Tamir developed in [30] a PTAS for the problem in fixed dimension. Our scheme in Section 2.1 includes a PTAS for the minimum R-dimensional binary MCK in fixed dimension.

In Section 2.2 we reduce unbounded DSP to the minimum (non-binary) *R-dimensional MCK*. Chandra et al. [3] gave a PTAS for the *maximum* version of this problem in fixed dimension; their scheme solves as a procedure a linear program. Our scheme yields the first combinatorial scheme for this problem.

Set Cover/Covering Integer Programs: As mentioned above, our problems include as a special case the multi-set multi-cover problem. Set cover and its generalizations have been extensively studied. (A comprehensive survey is given in [34].) Feige showed that in general set cover is hard to approximate within factor $\Omega(\log |E|)$, where E is the set of elements to be covered. This hardness result carries over to multi-set multi-cover. The best approximation ratio for set cover is $(1 + \ln |E|)$ [5]. For multi-set multi-cover, the best ratio is $O(\log \max_S |S|)$, where

⁴See in Section 3.

$|S|$ is the size of the multi-set S when counting elements with multiplicity [28]. This yields an $O(\log n)$ -approximation algorithm for general instances of DSP with unbounded supply, where $n = \sum_{j=1}^R n_j$.

Covering integer programs form a large subclass of integer programs encompassing such NP-hard problems as minimum knapsack and set cover. This implies the hardness of CIP in fixed dimension (i.e., where R is a fixed constant). For general instances, the hardness of approximation results for set cover carry over to CIP. Dobson [6] gave an algorithm that outputs a solution of cost $O(\max_{1 \leq i \leq n} \{\log(\sum_{j=1}^R a_{ji})\})$ times the integral optimum. It was unknown until the late 90's whether an $O(\log R)$ -approximation existed. Kolliopoulos and Young [20] settled this question. Their $O(\log R)$ -approximation yields the first constant approximation for CIP in fixed dimension. A comprehensive survey of other results is given in [27] (see also [19, 20]). The best known bounds for the CIP_∞ problem (that include existential improvements on the $O(\log R)$ factor) are due to Srinivasan ([32] and [31]). In this paper, we give the first approximation schemes for CIP and CIP_∞ in fixed dimension.

2 Deal Splitting with Packages

2.1 DSP with Bounded Multiplicity

2.1.1 Approximation Scheme

Suppose that the packages offered by each of the sellers have bounded multiplicity. Specifically, there are r_ℓ^i copies available from p_ℓ^i , $1 \leq \ell \leq N_i$. In this case, if $p_\ell^i = (n_{\ell 1}^i, \dots, n_{\ell R}^i)$, $1 \leq \ell \leq N_i$, then the number of units available from seller i of the j -th good is $T_{ji} = \sum_{\ell=1}^{N_i} n_{\ell j}^i r_\ell^i$, for $1 \leq j \leq R$, $1 \leq i \leq m$. We now develop a PTAS for these instances, assuming that R is fixed.

Reduction to the R-MMCK Problem: Assume that we know the optimal cost, C , for our instance, then we reduce our problem to the minimum R -dimensional binary multiple choice knapsack problem. Recall that for some $R \geq 1$, an instance of binary R-MMCK consists of a single R -dimensional knapsack, of size b_j in the j -th dimension, and N sets of items. Each item has an R -dimensional size and is associated with a cost. The goal is to pack a subset of items, by selecting at most one item from each set, such that the total size of the packed items in dimension j is at least b_j , $1 \leq j \leq R$, and the overall cost is minimized.

For given values of C and ε , we define an instance for R-MMCK, such that if there is an optimal solution for DSP with cost C , we can find a solution for the DSP instance, whose cost is at most $C(1 + \varepsilon)$. Note that C can be ‘guessed’ in polynomial time within factor $(1 + \varepsilon)$, using binary search over the range $(0, \sum_{i=1}^m \sum_{\ell=1}^{N_i} r_\ell^i c(p_\ell^i))$.

Formally, given the value of C , the parameter ε and a DSP instance with bounded multiplicity, we construct an R-MMCK instance in which the knapsack capacities in the R dimensions are $b_j = n_j$, $1 \leq j \leq R$. Also, we have $N = \sum_{i=1}^m N_i$ sets of items, denoted by A_ℓ^i , $1 \leq i \leq m$, $1 \leq \ell \leq N_i$. Let \hat{K}_ℓ^i be the integer value satisfying $r_\ell^i c(p_\ell^i) \in [\hat{K}_\ell^i \varepsilon C/N, (\hat{K}_\ell^i + 1) \varepsilon C/N]$, then the number of items in A_ℓ^i is $K_\ell^i = \min(\hat{K}_\ell^i, \lfloor N/\varepsilon \rfloor)$. The set A_ℓ^i represents a sale of the package p_ℓ^i

which partially fulfills the order. In particular, the k -th item in A_ℓ^i , denoted (i, ℓ, k) , represents a sale of $\gamma(i, \ell, k) \in \{1, 2, \dots, r_\ell^i\}$ copies of p_ℓ^i such that $c(i, \ell, k)$, the total cost incurred by these copies, is in $[k\varepsilon C/N, (k+1)\varepsilon C/N)$. This total cost is rounded down to the nearest integral multiple of $\varepsilon C/N$; thus, $c(i, \ell, k) = k\varepsilon C/N$. The size of the item (i, ℓ, k) in dimension j , $1 \leq j \leq R$, denoted by $s_j(i, \ell, k)$, is the total number of units of the j -th good that we can obtain, such that the total (rounded down) cost is $c(i, \ell, k)$, namely, $s_j(i, \ell, k) = \gamma(i, \ell, k)n_{\ell j}^i$.

Approximating the Optimal Solution for R-MMCK: Given an instance of R-MMCK, we ‘guess’ the set S of items of maximal costs in the optimal solution, where $|S| \leq h = \min(N, \lfloor \frac{2R(1+\varepsilon)}{\varepsilon} \rfloor)$. We choose the value of h such that the resulting solution is guaranteed to be within $1 + \varepsilon$ from the optimal, as computed below. Let $E(S)$ be the subset of items with costs that are larger than the minimal cost of any item in S , that is, $E(S) = \{(i, \ell, k) \notin S \mid c(i, \ell, k) > c_{\min}(S)\}$, where $c_{\min}(S) = \min_{(i, \ell, k) \in S} c(i, \ell, k)$. We select all the items $(i, \ell, k) \in S$, and eliminate from the instance all the items $(i, \ell, k) \in E(S)$ and the sets A_ℓ^i from which an item has been selected. In the next step we find an optimal *basic solution* for the following linear program, in which $x_{i, \ell, k}$ is an indicator variable for the selection of the item (i, ℓ, k) , $1 \leq i \leq m$, $1 \leq \ell \leq N_i$, $1 \leq k \leq K_\ell^i$.

$$\begin{aligned}
(LP(S)) \quad & \text{minimize} && \sum_{i=1}^m \sum_{\ell=1}^{N_i} \sum_{k=1}^{K_\ell^i} x_{i, \ell, k} \cdot c(i, \ell, k) \\
& \text{subject to :} && \sum_{k=1}^{K_\ell^i} x_{i, \ell, k} \leq 1 \text{ for } i = 1, \dots, m, \ell = 1, \dots, N_i \\
& && \sum_{i=1}^m \sum_{\ell=1}^{N_i} \sum_{k=1}^{K_\ell^i} s_j(i, \ell, k) \geq n_j \text{ for } j = 1, \dots, R \\
& && x_{i, \ell, k} = 1 \text{ for } (i, \ell, k) \in S \\
& && x_{i, \ell, k} = 0 \text{ for } (i, \ell, k) \in E(S) \\
& && 0 \leq x_{i, \ell, k} \leq 1 \text{ for } (i, \ell, k) \notin S \cup E(S)
\end{aligned}$$

Rounding the Fractional Solution: Given an optimal fractional solution for R-MMCK, we get an integral solution as follows. For any i , $1 \leq i \leq m$ and ℓ , $1 \leq \ell \leq N_i$ let $k_{\max} = k_{\max}(\ell, i)$ be the maximal value of $1 \leq k \leq K_\ell^i$ such that $x_{i, \ell, k} > 0$; then we set $x_{i, \ell, k_{\max}} = 1$ and, for any other item in A_ℓ^i , $x_{i, \ell, k} = 0$. Finally, we return to the DSP instance and take the maximum number of copies of the package p_ℓ^i whose total (rounded down) cost is $c(i, \ell, k_{\max})$.

2.1.2 Analysis

We use the next three lemmas to show that the scheme yields a $(1 + \varepsilon)$ -approximation to the optimum cost, and that the resulting integral solution is feasible.

Lemma 2.1 *If there exists an optimal (integral) solution for DSP with cost C , then the integral solution obtained from the rounding for R-MMCK has the cost $\hat{z} \leq (1 + \varepsilon)C$.*

Proof: Let \mathbf{x}^* be an optimal integral solution for the linear program of the R-MMCK instance, and let S^* be the corresponding subset of items, that is, $S^* = \{(i, \ell, k) \mid x_{i,\ell,k}^* = 1\}$. If $|S^*| < h$ then we are done (the scheme outputs a $(1 + \varepsilon)$ -approximation to the optimal cost: this is due to the initial guess of C); otherwise, let $S^* = \{(i_1, \ell_1, k_1), \dots, (i_g, \ell_g, k_g)\}$, such that $c(i_1, \ell_1, k_1) \geq \dots \geq c(i_g, \ell_g, k_g)$, for some $g > h$. Let $S_h^* = \{(i_1, \ell_1, k_1), \dots, (i_h, \ell_h, k_h)\}$, and $\sigma = \sum_{t=1}^h c(i_t, \ell_t, k_t)$. Then, for any item $(i, \ell, k) \notin (S_h^* \cup E(S_h^*))$, we have $c(i, \ell, k) \leq \sigma/h$. Let z^* , \hat{z} denote the optimal (integral) solution and the solution output by the scheme for the R-MMCK instance, respectively. We denote by $\mathbf{x}^B(S_h^*)$, $\mathbf{x}^I(S_h^*)$ the basic and integral solutions of $LP(S_h^*)$.

We note that, by our rounding technique, for any $1 \leq i \leq m$, $1 \leq \ell \leq N_i$, the cost of the item selected from A_ℓ^i is $c(i, \ell, k_{max})$. Hence, we get that

$$\begin{aligned} z^* &\geq \sum_{i=1}^m \sum_{\ell=1}^{N_i} \sum_{k=1}^{K_\ell^i} c(i, \ell, k) x_{i,\ell,k}^B(S_h^*) \\ &\geq \sum_{i=1}^m \sum_{\ell=1}^{N_i} \sum_{k=1}^{K_\ell^i} c(i, \ell, k) x_{i,\ell,k}^I(S_h^*) - \delta \geq \hat{z} - \delta, \end{aligned}$$

where $\delta = \sum_{(i,\ell,k) \in F} c(i, \ell, k)$, and F is the set of items for which the basic variable was a fraction, that is, $F = \{(i, \ell, k) \mid x_{i,\ell,k}^B(S_h^*) < 1\}$. This follows from the fact that we round up to 1 only $x_{i,\ell,k_{max}}$, for all $1 \leq i \leq m$ and $1 \leq \ell \leq N_i$.

Recall that in any *basic* solution for a linear program, the number of non-zero variables is bounded by the number of tight constraints (i.e., constraints that are satisfied with equality) in some optimal solution (since non-tight constraints can be omitted). Assume that in the optimal (fractional) solution of $LP(S_h^*)$ there are L tight constraints, where $0 \leq L \leq N + R$. Then in the basic solution $\mathbf{x}^B(S_h^*)$, at most L variables can be strictly positive. We note that the number of tight MCK constraints (i.e., constraints of the form $\sum_{k=1}^{K_\ell^i} x_{i,\ell,k} \leq 1$) is at least $L - R$. We claim that $|F| \leq 2R$. Indeed, assume by contradiction that $|F| > 2R$, then the number of integral variables (i.e., variables that are assigned the value ‘1’) is at most $L - 2R - 1$, and this is also the number of MCK constraints that become tight due to integral variables. Now, each pair of the remaining (non-integral) variables tightens a single MCK constraint. Since the number of additional constraints to be tightened is at least $R + 1$, we get additional $2(R + 1)$ fractional variables. It follows that the total number of strictly positive variables is at least $L - 2R - 1 + 2(R + 1) > L$.

Finally, we note that, for any $(i, \ell, k) \in F$ $c(i, \ell, k) \leq \sigma/h$, since $F \cap (S_h^* \cup E(S_h^*)) = \emptyset$. Hence, we get that $z^* \geq \hat{z} + \frac{2R\sigma}{h} \geq \hat{z} + \frac{2R\hat{z}}{h} \geq \hat{z}(1 - \frac{2R}{\lceil \frac{2R(1+\varepsilon)}{\varepsilon} \rceil}) \geq \frac{\hat{z}}{1+\varepsilon}$. ■

Lemma 2.2 *The scheme yields a feasible solution for the DSP instance.*

Proof: We need to show that (i) the number of copies obtained from p_ℓ^i is at most r_ℓ^i , and (ii) the number of units obtained from the j -th good is at least n_j , for all $1 \leq j \leq R$. We note that (i) follows from the definition of K_ℓ^i and the fact that $k_{max}(i, \ell) \leq K_\ell^i$. Also, (ii) holds since our rounding guarantees that $k_{max}(i, \ell)$, the number of copies obtained from each package in the integral solution, is at least the number obtained in the fractional solution. ■

Lemma 2.3 *The cost of the integral solution for the DSP instance is at most $\hat{z} + \varepsilon C$.*

Proof: We note that for any package p_ℓ^i , the cost incurred by the integral solution for the DSP instance is at most $c(i, \ell, k_{\max}(i, \ell)) + \varepsilon C/N$. It follows that the total cost value for the R-MMCK instance is increased at most by εC . ■

Combining the above lemmas we get:

Theorem 2.4 *There is a polynomial time approximation scheme for DSP instances with fixed number of goods and bounded multiplicity.*

Consider an instance of CIP in fixed dimension, R . We want to minimize $\sum_{i=1}^n c_i x_i$ subject to the constraints $\sum_{i=1}^n a_{ij} x_i \geq b_j$ for $j = 1, \dots, R$, and $x_i \in \{0, 1, \dots, d_i\}$ for $i = 1, \dots, n$. We can represent such a program as an instance of DSP with $m = n$ sellers, each offering a single package i of multiplicity d_i . The number of units required from the j -th good is $n_j = b_j$. The resulting scheme is simple and similar to the PTAS for d -dimensional knapsack [9].

Corollary 2.5 *The above is a PTAS for CIP in fixed dimension.*

2.1.3 Hardness Results

In this section we show that an FPTAS or even an EPTAS for CIP is unlikely to exist. Since CIP is a special case of DSP, the same hardness results hold also for DSP. Throughout this section we assume that $R = 2$ and $d_i = 1$ for all $1 \leq i \leq n$. The proofs of our results are similar to the proof of hardness for two-dimensional knapsack, as given in [22]. Thus, we omit some of the details.

Denote by $OPT(\mathcal{I})$ the value of an optimal solution for an instance \mathcal{I} of CIP. In deriving the hardness results, we use the following parameterized version of the *subset sum* problem, known as *sized subset sum*. Given a set of positive integers $L = \{x_1, \dots, x_n\}$, a positive integer S and an integer $k \geq 1$, decide if there is a subset $L' \subseteq L$ of size k , such that the sum of elements in L' is *exactly* S (in this case we say that the input is *satisfied*). The sized subset sum problem is known to be both *NP*-complete and *W[1]*-hard [7]. We give a reduction from an instance (L, S, k) of sized subset sum to an instance of CIP, denoted by $R(L, S, k)$.

Given an instance (L, S, k) , we first modify the values of elements in L . Define

$$\tilde{x}_i = \frac{x_i + \frac{k-1}{k} \cdot S}{k},$$

and let $\tilde{L} = \{\tilde{x}_1, \dots, \tilde{x}_n\}$. Note that for any $1 \leq i \leq n$, we have that $0 \leq \tilde{x}_i \leq \frac{2 \cdot S}{k}$ (w.l.o.g $x_i \leq S$).

Lemma 2.6 [22] *The instance (L, S, k) is satisfied if and only if (\tilde{L}, S, k) is satisfied.*

Now, we define the instance $R(L, S, k)$ of CIP. The items are $\{1, \dots, n\}$, where each item i has size $\bar{a}_i = (\tilde{x}_i, \frac{2 \cdot S}{k} - \tilde{x}_i)$ and cost $c_i = 1$. Also, the capacity of the bin is $b = (S, S)$. Note that $R(L, S, k)$ can be computed in polynomial time in the size of the instance (L, S, k) (and its size is also polynomial).

Lemma 2.7 *The instance (\tilde{L}, S, k) is satisfied if and only if $OPT(R(L, S, k)) \leq k$.*

Proof: If the instance (\tilde{L}, S, k) is satisfied then there is a subset $\{\tilde{x}_{i_1}, \dots, \tilde{x}_{i_k}\} = \tilde{L}' \subseteq \tilde{L}$ such that $\sum_{j=1}^k \tilde{x}_{i_j} = S$. Then the solution $A = \{a_{i_1}, \dots, a_{i_k}\}$ for $R(L, S, k)$ is feasible in both dimensions, i.e., $\sum_{j=1}^k s_{i_j,1} = \sum_{j=1}^k \tilde{x}_{i_j} = S$, and also $\sum_{j=1}^k s_{i_j,2} = \sum_{j=1}^k \left(\frac{2S}{k} - \tilde{x}_{i_j}\right) = S$. The cost of this solution is k , therefore $OPT(R(L, S, k)) \leq k$.

If $OPT(R(L, S, k)) \leq k$ then let $A = \{a_{i_1}, \dots, a_{i_\ell}\}$ be an optimal solution, where $\ell \leq k$. Since this is a valid solution, we have that

$$S \leq \sum_{j=1}^{\ell} a_{i_j,1} = \sum_{j=1}^{\ell} \tilde{x}_{i_j}, \quad (1)$$

and also

$$S \leq \sum_{j=1}^{\ell} a_{i_j,2} = \sum_{j=1}^{\ell} \left(\frac{2S}{k} - \tilde{x}_{i_j}\right) = 2 \cdot S \cdot \frac{\ell}{k} - \sum_{j=1}^{\ell} \tilde{x}_{i_j} \leq 2 \cdot S - \sum_{j=1}^{\ell} \tilde{x}_{i_j}. \quad (2)$$

Combining (1) and (2) we get that $S = \sum_{j=1}^{\ell} \tilde{x}_{i_j}$, thus (\tilde{L}, S, k) is satisfied. ■

By Lemmas 2.6 and 2.7, we have

Lemma 2.8 *For any instance (L, S, k) of sized subset sum, (L, S, k) is satisfied if and only if $OPT(R(L, S, k)) \leq k$.*

Suppose there exists an efficient approximation scheme $\mathcal{A}(\mathcal{I}, \varepsilon)$ for CIP. We now show how \mathcal{A} can be used to decide if an input for sized subset sum is satisfied.

Lemma 2.9 *Suppose that, for any $\varepsilon > 0$, there is an approximation scheme $\mathcal{A}(\mathcal{I}, \varepsilon)$ whose running time $f(1/\varepsilon) \cdot |\mathcal{I}|^{O(1)}$. Then, for any $k \geq 1$, there is an algorithm for sized subset sum with running time $f(2k) \cdot |(L, S, k)|^{O(1)}$.*

Proof: Consider the following algorithm for sized subset sum. Given an instance (L, S, k) , define the input $\mathcal{I} = R(L, S, k)$ for CIP, and run $\mathcal{A}(\mathcal{I}, \frac{1}{2k})$. If the optimal solution output by the algorithm is of value at most k return that (L, S, k) is satisfied, otherwise return that it cannot be satisfied.

Note that if $OPT(\mathcal{I}) \leq k$, the value output by \mathcal{A} is at most $\left(1 + \frac{1}{2k}\right)k = k + \frac{1}{2} < k + 1$. On the other hand, if $OPT(\mathcal{I}) > k$, the output value is at least $k + 1$. Also, by Lemma 2.8, (L, S, k) is satisfied if and only if $OPT(\mathcal{I}) \leq k$. Hence, the algorithm decides correctly if (L, S, k) is satisfied.

The construction of \mathcal{I} takes polynomial time in $|(L, S, k)|$, and running \mathcal{A} on the instance \mathcal{I} requires $f(2k) \cdot |R(L, S, k)|^{O(1)}$ steps. Thus, the running time of the algorithm is $f(2k) \cdot |(L, S, k)|^{O(1)}$. ■

We summarize the above discussion with the following results.

Theorem 2.10 *There is no FTPAS for CIP with $R = 2$ unless $P = NP$.*

Proof: Assume that, for any $\varepsilon > 0$, there is an approximation scheme $\mathcal{A}(\mathcal{I}, \varepsilon)$ for CIP with running time $f(1/\varepsilon) \cdot |\mathcal{I}|^{O(1)}$, where f is a polynomial. Then, by Lemma 2.9, for any $k \geq 1$, there is an algorithm for sized subset sum whose running time is $f(2k) \cdot |(L, S, k)|^{O(1)}$. Since f is polynomial and k is also polynomial in the input size, we obtained a polynomial time algorithm for sized subset sum, implying that $P = NP$. ■

Theorem 2.11 *There is no EPTAS for CIP with $R = 2$ unless $W[1] = FPT$.*

Proof: Similar to the proof of Theorem 2.10, assume that there is an *EPTAS* for two-dimensional knapsack. That is, there exists an algorithm $\mathcal{A}(\mathcal{I}, \varepsilon)$ that, given an instance \mathcal{I} for the problem, returns a $(1 - \varepsilon)$ -approximation for the optimal solution in $f(1/\varepsilon) \cdot |\mathcal{I}|^c$ steps. Then, by Lemma 2.9, there is an algorithm for sized subset sum whose running time is $f(2k) \cdot |(L, S, k)|^{c'}$. It follows that sized subset sum is *fixed parameter tractable*, which cannot hold unless $W[1] = FPT$. ■

Remark: The above proofs follow with slight modifications the steps of the proof of hardness in [22] for R -dimensional knapsack. This may suggest that there exists a reduction from CIP to R -dimensional knapsack, which implies Theorems 2.10 and 2.11, based on the known hardness results for R -dimensional knapsack. We leave this question open.

2.2 Unbounded DSP

Consider now the special case where the sellers have unbounded supply from each of the goods. As before, we formulate our problem as a linear program, however, instead of applying standard techniques to solve this program, we use a fast combinatorial approximation scheme of [11] to get a fractional solution that is within factor of $(1 + \varepsilon)$ from the optimal; then, we round the solution to obtain an integral solution that is close to the optimal.

2.2.1 Overview of the Scheme

Our scheme, called *multi-dimensional cover with parameter ε* (MDC_ε), proceeds in the following steps.

- (i) For a given $\varepsilon \in (0, 1)$, let

$$\delta = \lceil R \cdot ((1/\varepsilon) - 1) \rceil. \quad (3)$$

- (ii) Recall that $N = \sum_{i=1}^m N_i$ is the total number of packages. We number the packages by $1, \dots, N$, such that $c_1 \geq c_2 \geq \dots \geq c_N$, where c_i is the cost of package i ; then, a_{ji} is the number of units of item j in package i , $1 \leq i \leq N$.

- (iii) Denote by Ω the set of integer vectors $\mathbf{x} = (x_1, \dots, x_N)$ satisfying $x_i \geq 0$ and $\sum_{i=1}^N x_i \leq \delta$. For any vector $\mathbf{x} \in \Omega$:

- Let $d \geq 1$ be the maximal integer i for which $x_i \neq 0$. Find a $(1 + \varepsilon)$ -approximation to the optimal (fractional) solution of the following linear program.⁵

$$(LP') \quad \text{minimize} \quad \sum_{i=d+1}^N c_i z_i$$

⁵See Lemma 2.13.

$$\text{subject to : } \begin{aligned} \sum_{i=d+1}^N a_{ji}z_i &\geq n_j - \sum_{i=1}^d a_{ji}x_i \text{ for } j = 1, \dots, R \\ z_i &\geq 0, \text{ for } i = d+1, \dots, N \end{aligned} \quad (4)$$

The constraints (4) reflect the fact that we need to obtain from each of the goods at least $n_j - \sum_{i=1}^d a_{ji}x_i$ units once we obtained the vector \mathbf{x} .

- Let \hat{z}_i , $d+1 \leq i \leq N$ be a $(1 + \varepsilon)$ -approximate solution for LP' . We take $\lceil \hat{z}_i \rceil$ as the integral solution. Denote by $C_{MDC}(\mathbf{x}) = \sum_{i=d+1}^N c_i \lceil \hat{z}_i \rceil$ the value obtained from the rounded solution, and let $c(\mathbf{x}) = \sum_{i=1}^N c_i x_i$.

(iv) Select the vector \mathbf{x} for which $C_{MDC_\varepsilon}(\mathbf{x}) = \min_{\mathbf{x}}(c(\mathbf{x}) + C_{MDC}(\mathbf{x}))$.

2.2.2 Analysis

We now show that MDC_ε is a PTAS for DSP with unbounded supply. Let C_o be the optimal cost for DSP (in which we take an *integral* number of units from each package).

Theorem 2.12 (i) If $C_o \neq 0, \infty$ then $C_{MDC_\varepsilon}/C_o < 1 + \varepsilon$. (ii) The running time of algorithm MDC_ε is $O(N^{\lceil 2R/\varepsilon \rceil} \cdot \frac{1}{\varepsilon^2} \log(\max(C, n_{max})))$, where

$$C = \max_{1 \leq i \leq N} c_i \quad (5)$$

is the maximal cost of any package, and $n_{max} = \max_{1 \leq j \leq R} n_j$ is the maximal number of units required from any of the goods.

We use in the proof the next results. Let LP_1 be a linear program of the form

$$(LP_1) \quad \begin{aligned} &\text{minimize} && \sum_{i=1}^m c_i z_i \\ &\text{subject to :} && \sum_{i=1}^m a_{ji} z_i \geq b_j \text{ for } j = 1, \dots, R \end{aligned} \quad (6)$$

$$z_i \leq u_i, \text{ for } i = 1, \dots, m \quad (7)$$

$$z_i \geq 0, \text{ for } i = 1, \dots, m$$

The next lemma is due to [11].

Lemma 2.13 For any $\varepsilon > 0$, a $(1 + \varepsilon)$ -approximation to the optimal (fractional) solution for LP_1 can be found in $O(\varepsilon^{-2} m \log(\mathbf{c}^T \mathbf{u}))$ oracle calls, where $\mathbf{c} = (c_1, \dots, c_m)$ and $\mathbf{u} = (u_1, \dots, u_m)$.

We note that for a system of inequalities as given in LP' there is a solution in which at most R variables get non-zero values. This follows from the fact that the number of non-trivial constraints is R . Hence, it suffices to solve LP' for the $\binom{N-d}{R}$ possible subsets of R variables, out of (z_{d+1}, \dots, z_N) . This can be done in polynomial time since R is a fixed constant.

Now, for any $d+1 \leq i \leq N$, we can upper bound z_i by n_{max} . Therefore, we can formulate LP' as LP_1 , where $m = R$ with $u_i = n_{max}$ for all i , and using Lemma 2.13 we get

Corollary 2.14 A $(1 + \varepsilon)$ -approximation to the optimal solution for LP' can be found in $O(\varepsilon^{-2}R \log(Cn_{max}))$ oracle calls, where C is defined in (5).

Proof of Theorem 2.12: For showing (i), assume that the optimal (integral) solution for the DSP instance is obtained by the vector $\mathbf{y} = (y_1, \dots, y_N)$. If $\sum_{i=1}^N y_i \leq \delta$ then $C_{MDC_\varepsilon} = C_o$, since in this case \mathbf{y} is a valid solution and $\mathbf{y} \in \Omega$, therefore, in some iteration MDC_ε will examine \mathbf{y} .

Suppose that $\sum_{i=1}^N y_i > \delta$, and let d be the value satisfying $\sum_{i=1}^{d-1} y_i < \delta$, $\sum_{i=1}^d y_i \geq \delta$, then we define the vector $\mathbf{x} = (y_1, \dots, y_{d-1}, x_d, 0, \dots, 0)$, such that $y_1 + \dots + y_{d-1} + x_d = \delta$. (Note that $x_d \neq 0$.) Let $\tilde{C}_o(\mathbf{x}) = \sum_{i=d+1}^N c_i \hat{z}_i$ be the approximate fractional solution for LP' . We note that $\mathbf{x} \in \Omega$; also, since $c_i \leq c_d$ for all $d+1 \leq i \leq N$ and at most R of the z_i values are strictly positive, we get that

$$C_{MDC}(\mathbf{x}) - \tilde{C}_o(\mathbf{x}) \leq Rc_d. \quad (8)$$

Let $C_o(\mathbf{x})$ be the cost of an optimal fractional solution for LP' with the vector \mathbf{x} . Note that C_o , the cost of an optimal (integral) solution for DSP, satisfies $C_o > c(\mathbf{x}) + C_o(\mathbf{x})$, since $C_o(\mathbf{x})$ is a lower bound for the cost incurred by the integral values y_{d+1}, \dots, y_N . In addition, $c(\mathbf{x}) + C_{MDC}(\mathbf{x}) \geq C_{MDC_\varepsilon}$. Hence, we get that

$$\begin{aligned} \frac{C_o}{C_{MDC_\varepsilon}} &\geq \frac{c(\mathbf{x}) + C_o(\mathbf{x})}{c(\mathbf{x}) + C_{MDC}(\mathbf{x})} \\ &> 1 - \frac{C_{MDC}(\mathbf{x}) - C_o(\mathbf{x})}{c(\mathbf{x}) + C_{MDC}(\mathbf{x}) - C_o(\mathbf{x})} \\ &\geq 1 - \frac{C_{MDC}(\mathbf{x}) - \tilde{C}_o(\mathbf{x})(1 - \varepsilon)}{c(\mathbf{x}) + C_{MDC}(\mathbf{x}) - \tilde{C}_o(\mathbf{x})} \\ &\geq (1 - \varepsilon) \left(1 - \frac{C_{MDC}(\mathbf{x}) - \tilde{C}_o(\mathbf{x})}{c(\mathbf{x}) + C_{MDC}(\mathbf{x}) - \tilde{C}_o(\mathbf{x})}\right) \\ &\geq (1 - \varepsilon) \left(1 - \frac{C_{MDC}(\mathbf{x}) - \tilde{C}_o(\mathbf{x})}{\delta c_d + C_{MDC}(\mathbf{x}) - \tilde{C}_o(\mathbf{x})}\right) \end{aligned}$$

The third inequality follows from the fact that $\tilde{C}_o(\mathbf{x})(1 - \varepsilon) \leq C_o(\mathbf{x}) \leq \tilde{C}_o(\mathbf{x})$, and the last inequality follows from the fact that $c(\mathbf{x}) \geq \delta c_d$.

Let $f(w) = w/(a + w)$, and define $w = C_{MDC}(\mathbf{x}) - \tilde{C}_o(\mathbf{x})$, $a = \delta c_d$; then, letting $h = Rc_d$, we have that

$$1 - \frac{w}{a + w} \geq 1 - f(h) \geq 1 - \varepsilon.$$

The first inequality follows from (8) and the fact that $f(w)$ is monotone increasing; the second inequality follows from (3).

Thus, we get that $C_o/C_{MDC_\varepsilon} \geq (1 - \varepsilon)^2$. By taking in the scheme $\tilde{\varepsilon} = \varepsilon/2$ we get the statement of the theorem.

Next, we show (ii). Note that $|\Omega| = O(N^\delta)$, since the number of possible choices of N non-negative integers, whose sum is at most δ is bounded by $\binom{N+\delta}{\delta}$. Now, given a vector $\mathbf{x} \in \Omega$, we can compute $C_{MDC}(\mathbf{x})$ in $O(N^R)$ steps, since at most R variables out of z_{d+1}, \dots, z_N can have non-zero values. Multiplying by the complexity of the FPTAS for fractional covering, as given in Corollary 2.14, we get the statement of the theorem. \blacksquare

Recall that DSP with unbounded supply is equivalent to CIP_∞ . Hence, we have

Theorem 2.15 *There is a PTAS for CIP_∞ with n variables and fixed dimension, R , whose running time is $O(N^{\lceil 2R/\varepsilon \rceil} \cdot \frac{1}{\varepsilon^2} \log(\max(C, n_{max}))$)*

3 Approximation Scheme for Deal Splitting with Price Tables

We now describe a PTAS for DS with price tables and fixed number of goods. Our scheme applies to any instance of DST satisfying the following properties. **(P1) Volume discount.** If we increase the quantity bought from each of the goods, the unit cost can only decrease; that is, let $(a_1^1, \dots, a_R^1), (a_1^2, \dots, a_R^2)$ be two vectors representing feasible sales for S_i (i.e., out of the entries in the price table of S_i), for some $1 \leq i \leq m$. If $a_j^2 \geq a_j^1$ for all $1 \leq j \leq R$, then the unit costs corresponding to the two vectors satisfy $c_j^2 \leq c_j^1$ for all j . **(P2) Dominance.** If the vectors $(d_1^1, \dots, d_R^1), (d_1^2, \dots, d_R^2)$ represent valid sales (vis a vis the price table) for S_i , then the vector $\max((d_1^1, \dots, d_R^1), (d_1^2, \dots, d_R^2))$ also represents a valid sale for S_i , where the maximum is taken coordinate-wise. Table 1 satisfies the volume discount and the dominance properties.

We note that the properties (P1) and (P2) are quite reasonable in commercial scenarios, reflecting the desire of each seller to increase its part in the deal, by selling more units from each of the goods. (P1) implies that as the quantities increase, the unit prices decrease; (P2) allows for more combinations of the goods for the buyer. It can be shown (by reduction from Partition) that DST is NP-hard even for instances that satisfy properties (P1) and (P2), already for $R = 1$.

Assume that we know the optimal cost, C , for our instance. Then, for a given value of $\varepsilon > 0$, we define an instance of R-MMCK, whose optimal solution induces a solution for DST with cost at most $(1 + \varepsilon)C$. We then find an optimal fractional solution for the R-MMCK instance. This gives an almost optimal fractional solution for the DST instance. Finally, we use non-standard rounding to obtain an integral solution whose cost is within factor $(1 + \varepsilon)$ from the fractional solution. Note that C can be ‘guessed’ in polynomial time within factor $(1 + \varepsilon)$, using binary search over the range $(0, mR \cdot \max_{i,j} \max_{1 \leq \ell \leq m_i} u_{\ell j} c_{\ell j})$, i.e., we allow to take the maximum number of units from the j -th good in the ℓ th range, for $1 \leq \ell \leq m_i, 1 \leq i \leq m, 1 \leq j \leq R$.

Reduction to the R-MMCK Problem: Given the value of C , the parameter ε and a DST instance with m price tables, we construct an R-MMCK instance which consists of a single R -dimensional knapsack with capacities $b_j = n_j, 1 \leq j \leq R$, and m sets of items; each set A_i has $m_i \cdot \lfloor m/\varepsilon \rfloor^R$ items, $1 \leq i \leq m$. Each of the items in A_i represents a sale of the i -th seller, which (partially) satisfies the order. Specifically, each item in A_i is an integer vector $(i, \ell, k_1, \dots, k_R)$, where ℓ is the range in the i -th price table from which we choose the goods, and $0 \leq k_j \leq \lfloor m/\varepsilon \rfloor$ is the contribution of the j -th good, bought from the i -th seller, to the total cost. We take this contribution as an integral multiple of $\varepsilon C/m$; for each vector we find the maximal

number of units of each good that can be bought with this vector. If for some integer $g \geq 1$, $k_j \varepsilon C/m < g c_{\ell_j} \leq (k_j+1) \varepsilon C/m$ then we buy g units from the j -th good and round down the cost to $k_j \varepsilon C/m$. The cost of an item $(i, \ell, k_1, \dots, k_R)$ in A_i is given by $c(i, \ell, k_1, \dots, k_R) = \varepsilon C/m \sum_{j=1}^R k_j$. We denote by $s_j(i, \ell, k_1, \dots, k_R)$ the maximum total number of units of the j -th good that can be bought from S_i in the ℓ -th range of its price table, at the cost $k_j \varepsilon C/m$, $1 \leq j \leq R$.

Approximating the Optimal Solution for R-MMCK: Given an instance of R-MMCK, we ‘guess’ the set S of items of maximal costs in the optimal solution, where $|S| \leq h = \min(m, \lfloor \frac{2R(1+\varepsilon)}{\varepsilon} \rfloor)$. Let $E(S)$ be the subset of items with costs that are larger than the minimal cost of any item in S , that is, $E(S) = \{(i, \ell, k_1, \dots, k_R) \notin S \mid c(i, \ell, k_1, \dots, k_R) > c_{\min}(S)\}$, where $c_{\min}(S) = \min_{(i, \ell, k_1, \dots, k_R) \in S} c(i, \ell, k_1, \dots, k_R)$.

We select all the items $(i, \ell, k_1, \dots, k_R) \in S$ and indicate that items $(i, \ell, k_1, \dots, k_R) \in E(S)$ cannot be selected. In the next step we find an optimal *basic solution* for the following linear program, in which $x_{i, \ell, k_1, \dots, k_R}$ is an indicator variable for the selection of an item $(i, \ell, k_1, \dots, k_R)$.

$$\begin{aligned}
(LP(S)) \quad & \text{minimize} && \sum_{i=1}^m \sum_{\ell=1}^{m_i} \sum_{k_1, \dots, k_R} c(i, \ell, k_1, \dots, k_R) x_{i, \ell, k_1, \dots, k_R} \\
& \text{subject to :} && \sum_{\ell=1}^{m_i} \sum_{k_1, \dots, k_R} x_{i, \ell, k_1, \dots, k_R} \leq 1 \text{ for } i = 1, \dots, m \\
& && \sum_{i=1}^m \sum_{\ell=1}^{m_i} \sum_{k_1, \dots, k_R} s_j(i, \ell, k_1, \dots, k_R) x_{i, \ell, k_1, \dots, k_R} \geq n_j \text{ for } j = 1, \dots, R \\
& && x_{i, \ell, k_1, \dots, k_R} = 1 \text{ for } (i, \ell, k_1, \dots, k_R) \in S \\
& && x_{i, \ell, k_1, \dots, k_R} = 0 \text{ for } (i, \ell, k_1, \dots, k_R) \in E(S) \\
& && 0 \leq x_{i, \ell, k_1, \dots, k_R} \leq 1 \text{ for } (i, \ell, k_1, \dots, k_R) \notin S \cup E(S)
\end{aligned}$$

Rounding the Fractional Solution: Given an optimal fractional solution for R-MMCK, we now return to the DST instance and get an integral solution as follows. Suppose that we have $D = D(i)$ fractional variables for some set A_i , $x_{i, \ell_1, k_{11}, \dots, k_{1R}}, \dots, x_{i, \ell_D, k_{D1}, \dots, k_{DR}}$, then we buy from the i -th seller $\max_{1 \leq d \leq D} s_j(i, \ell_d, k_{d1}, \dots, k_{dR})$ units of the j -th good, $1 \leq j \leq R$.

3.1 Analysis

We now show that the above scheme yields a $(1 + \varepsilon)$ -approximation to the optimum cost for DST, and that the resulting (integral) solution is feasible. We prove three lemmas.

Lemma 3.1 *If there exists an optimal (fractional) solution with cost C for the R-MMCK instance, then there exists a (fractional) solution with cost at most $(1 + \varepsilon)C$ for the DST instance.*

Proof: For any $\varepsilon' > 0$, in any fractional solution for R-MMCK with ε' , the cost of each of the selected items $(i, \ell, k_1, \dots, k_R)$ in the DST instance is at most $(c(i, \ell, k_1, \dots, k_R) + R\varepsilon' C/m) x_{i, \ell, k_1, \dots, k_R}$. Since for all $1 \leq i \leq m$ $\sum_{\ell=1}^{m_i} \sum_{k_1, \dots, k_R} x_{i, \ell, k_1, \dots, k_R} \leq 1$, this yields an increase of at most $R\varepsilon' C/m$ for the seller S_i . By taking $\varepsilon' = \varepsilon/R$, we get that the overall increase in the cost is $R\varepsilon' C = \varepsilon C$. ■

Lemma 3.2 *The integral solution obtained from the fractional solution for LP(S) yields a ratio of at most $(1 + \varepsilon)$ to the optimal cost for the DST instance.*

Proof: Let \mathbf{x}^* be an optimal integral solution for the linear program for the R-MMCK instance, and let $S^* = \{(i, \ell, k_1, \dots, k_R) \mid x_{i, \ell, k_1, \dots, k_R}^* = 1\}$ be the corresponding subset of items. As in the proof of Lemma 2.1, we may assume that $|S^*| \geq h$, otherwise we are done. Let

$$S^* = \{(i_1, \ell_1, k_{11}, \dots, k_{1R}), \dots, (i_r, \ell_r, k_{r1}, \dots, k_{rR})\},$$

such that $c(i_1, \ell_1, k_{11}, \dots, k_{1R}) \geq \dots \geq c(i_r, \ell_r, k_{r1}, \dots, k_{rR})$, for some $r > h$, and let

$$S_h^* = \{(i_1, \ell_1, k_{11}, \dots, k_{1R}), \dots, (i_h, \ell_h, k_{h1}, \dots, k_{hR})\}.$$

We define $\sigma = \sum_{t=1}^h c(i_t, \ell_t, k_{t1}, \dots, k_{tR})$ to be the total cost of the items in S_h^* . Then, for any item $(i, \ell, k_1, \dots, k_R) \notin (S_h^* \cup E(S_h^*))$, we have $c(i, \ell, k_1, \dots, k_R) \leq \sigma/h$.

We use below the notation $s_j(i, d)$ when referring to $s_j(i, \ell_d, k_{d1}, \dots, k_{dR})$. Also, we refer throughout the discussion to the solution for the DST instance, as obtained by our rounding procedure.

Let $c(\max_{1 \leq d \leq D} s_j(i, d))$ be the total cost of buying the j -th good in the entry of the i th price table where we obtain $\max_{1 \leq d \leq D} s_j(i, d)$ units from good j , $1 \leq j \leq R$. The heart of the proof is the following claim.

Claim 3.3 *For any $1 \leq i \leq m$, the cost of buying from the i -th seller satisfies*

$$\sum_{j=1}^R c(\max_{1 \leq d \leq D} s_j(i, d)) \leq \sum_{d=1}^D c(i, \ell_d, k_{d1}, \dots, k_{dR}).$$

Proof: Note that, by our rounding technique, the vector giving the amounts bought from S_i from each of the goods satisfies

$$(\max_{1 \leq d \leq D} s_1(i, d), \dots, \max_{1 \leq d \leq D} s_R(i, d)) \geq (s_1(i, \ell_d, k_{d1}, \dots, k_{dR}), \dots, s_R(i, \ell_d, k_{d1}, \dots, k_{dR})),$$

for all $1 \leq d \leq D$. By the volume discount property, the total cost of the rounded solution satisfies $c(\max_{1 \leq d \leq D} s_1(i, d), \dots, \max_{1 \leq d \leq D} s_R(i, d)) \leq \sum_{d=1}^D c(i, \ell_d, k_{d1}, \dots, k_{dR})$. ■

Let z^* denote the optimal (integral) solution for the R-MMCK instance. Denote by $\mathbf{x}^B(S_h^*)$ a basic solution for LP(S_h^*), and let $\mathbf{x}^I(S_h^*)$ be an integral solution obtained by setting $x_{i, \ell_d, k_{d1}, \dots, k_{dR}} = 1$ for all $1 \leq d \leq D$. From Claim 3.3, we can bound the total cost of the solution output by the scheme by comparing z^* to the cost of $\mathbf{x}^I(S_h^*)$. In particular,

$$\begin{aligned} z^* &\geq \sum_{i=1}^m \sum_{\ell=1}^{m_i} \sum_{k_1, \dots, k_R} c(i, \ell, k_1, \dots, k_R) x_{i, \ell, k_1, \dots, k_R}^B(S_h^*) \\ &\geq \sum_{i=1}^m \sum_{\ell=1}^{m_i} \sum_{k_1, \dots, k_R} c(i, \ell, k_1, \dots, k_R) x_{i, \ell, k_1, \dots, k_R}^I(S_h^*) - \delta \end{aligned}$$

where $\delta = \sum_{(i, \ell, k_1, \dots, k_R) \in F} c(i, \ell, k_1, \dots, k_R)$, and F is the set of items for which the basic variable was a fraction, that is, $F = \{(i, \ell, k_1, \dots, k_R) \mid x_{i, \ell, k_1, \dots, k_R}^B(S_h^*) < 1\}$

Assume that in the optimal (fractional) solution of $LP(S_h^*)$ there are L tight constraints, where $0 \leq L \leq m + R$, then in the basic solution $\mathbf{x}^B(S_h^*)$, at most L variables can be strictly positive. Thus, at least $L - 2R$ variables get an integral value (i.e. ‘1’), and $|F| \leq 2R$. By an argument similar to the argument in the proof of Lemma 2.1, we get the statement of the lemma.

Now, we note that, from Lemma 3.1, we have $(1 + \varepsilon)^2$ -approximation for DST, and since C is guessed within factor $1 + \varepsilon$, we get a $(1 + \varepsilon)^3$ -approximation. By taking $\varepsilon' = \varepsilon/4$ we get the statement of the lemma. ■

Lemma 3.4 *The integral solution obtained by the rounding is feasible for DST.*

Proof: Given the set of items with fractional variables in A_i , recall that we buy from S_i $\max_{1 \leq d \leq D} s_j(i, d)$ units from the j -th good, $1 \leq j \leq R$. By property (P2), there exists an entry in the price table of seller i , in which the number of units available from the j th good is $\max_{1 \leq d \leq D} s_j(i, d)$. This entry guarantees that the covering constraint is satisfied for all goods, that is, we buy at least n_j units of the j th good, and the stock constraints of the seller are not violated. ■

Combining the above lemmas we get:

Theorem 3.5 *There is a polynomial time approximation scheme for any DST instance satisfying properties (P1) and (P2), with fixed number of goods.*

References

- [1] M. Bichler, J. Kalagnanam, H.S. Lee, J. Lee, “Winner Determination Algorithms for Electronic Auctions: A Framework Design”. *EC-Web* 2002: 37–46.
- [2] G.J. Burke, J.E. Carrillo and A.J. Vakharia, “Heuristics for Sourcing from Multiple Suppliers with Alternative Quantity Discounts”. *European Journal of Operational Research* 186(1): 317–329, 2008.
- [3] A.K. Chandra, D.S. Hirschberg and C.K. Wong, “Approximate Algorithms for Some Generalized Knapsack Problems”. *Theoretical Computer Science* 3, pp. 293–304, 1976.
- [4] C. Chekuri and S. Khanna, “A PTAS for the Multiple Knapsack Problem”. In *Proc. of SODA*, pp. 213–222, 2000.
- [5] V. Chvátal, V. “A Greedy Heuristic for the Set Covering Problem”, *Math. Oper. Res.* 4, 1979, 233–235.
- [6] G. Dobson, “Worst-case Analysis of Greedy for Integer Programming with Nonnegative Data”. *Math. of Operations Research*, 7, pp. 515–531, 1982.
- [7] R. G. Downey and M. R. Fellows, “Fixed-parameter tractability and completeness II: On completeness for $W[1]$ ”. *Theoretical Computer Science*, 141(1-2): 109 – 131,1995.
- [8] R. G. Downey and M. R. Fellows, “Parameterized Complexity”. *Monographs in Computer Science*, 1999, Springer.

- [9] A. M. Frieze and M.R.B. Clarke, Approximation Algorithms for the m-dimensional 0-1 knapsack problem: worst-case and probabilistic analyses. In *European J. of Operational Research*, 15(1):100–109, 1984.
- [10] U. Feige. “A threshold of $\ln n$ for approximating set cover”. In *Proc. of 28th Symposium on Theory of Computing*, pp. 314–318, 1996.
- [11] L. Fleischer, “A Fast Approximation Scheme for Fractional Covering Problems with Variable Upper Bounds”. In *Proc. of the 15th ACM-SIAM Symposium on Discrete Algorithm*, pp. 994–1003, 2004.
- [12] T. Ghasemi and M. Razzazia. “Development of core to solve the multidimensional multiple-choice knapsack problem”. *Computers & Industrial Eng.*, Vol. 60 (2), pp. 349–360, 2011.
- [13] D. R. Goossens, A. J. T. Maas, F. C. R. Spieksma and J. van de Klundert. “Exact algorithms for procurement problems under a total quantity discount structure”. *European J. of Operational Research*, 178(2): 603–626, 2007.
- [14] B. Han, J. Leblet and G. Simon. “Hard multidimensional multiple choice knapsack problems, an empirical study”. *Computers & OR* 37(1), pp. 172–181, 2010.
- [15] He, Huang , Robert J., Kauffman , Hongyan, Xu , Lan, Zhao. “Mechanism design for e-procurement auctions: on the efficacy of post-auction negotiation and quality effort incentives”. *Electronic Commerce Research and Applications*, in Press, 2010.
- [16] T. Ibaraki, “Approximate algorithms for the multiple-choice continuous knapsack problems”, *J. of Operations Research Society of Japan*, 23, 28–62, 1980.
- [17] T. Ibaraki, T. Hasegawa, K. Teranaka and J. Iwase. “ The Multiple Choice Knapsack Problem”. *J. Oper. Res. Soc. Japan* 21, 59–94, 1978.
- [18] T. Ito, H. Hattori and M. Klein. “Multi-issue Negotiation Protocol for Agents: Exploring Nonlinear Utility Spaces”. *Proc. of the 20th International Joint Conference on Artificial Intelligence*, pp. 1347–1352, 2007.
- [19] S. G. Kolliopoulos, “Approximating covering integer programs with multiplicity constraints”, *Discrete Applied Math.*, 129:2–3, 461–473, 2003.
- [20] S. G. Kolliopoulos and N. E. Young, “Tight Approximation Results for General Covering Integer Programs”. In *Proc. of FOCS*, 522–528, 2001.
- [21] A. Kothari, D. Parkes and S. Suri, “Approximately-Strategyproof and Tractable Multi-Unit Auctions”. *Decision Support Systems* 39(1): 105–121, 2005.
- [22] A. Kulik and, H. Shachnai, “There is no EPTAS for two-dimensional knapsack”, *Information Processing Letters* 110(16): 707-710, 2010.
- [23] G.S. Lueker, “Two NP-complete problems in nonnegative integer programming”. Report # 178, Computer science Lab., Princeton Univ., 1975.

- [24] I. Marsa-Maestre, M.A. Lopez-Carmona, J.R. Velasco and E. de la Hoz. “Effective Bidding and Deal Identification for Negotiations in Highly Nonlinear Scenarios”. *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems*, (2) pp. 1057–1064, 2009.
- [25] R. Mohammad Ebrahim, J. Razmi and H. Haleh. “Scatter Search Algorithm for Supplier Selection and Order Lot Sizing under Multiple Price Discount Environment. *Advances in Engineering Software* 40(9), pp. 766–776, 2009.
- [26] D. H. Parente, “Best Practices for Online Procurement Auctions”. IGI-Global, 2008.
- [27] D. Pritchard. “Approximability of Sparse Integer Programs”. In *Proc. of ESA*, pp. 83–94, 2009.
- [28] S. Rajagopalan and V. V. Vazirani, “Primal-Dual RNC Approximation Algorithms for Set Cover and Covering Integer Programs”. *SIAM J. Comput.*, 28(2) pp. 525–540, 1998.
- [29] T. Sandholm, S. Suri, A. Gilpin and D. Levine, “ Winner determination in combinatorial auction generalizations”. In *Proc. of AAMAS*, 69–76, 2002.
- [30] H. Shachnai and T. Tamir, “Approximation Schemes for Generalized 2-dimensional Vector Packing with Application to Data Placement”. In *proc of Random-APPROX*, 2003.
- [31] A. Srinivasan, “An Extension of the Lovász Local Lemma, and its Applications to Integer Programming”. In *Proc. of SODA*, 6–15, 1996.
- [32] A. Srinivasan, “Improved Approximation Guarantees for Packing and Covering Integer Programs”. *SIAM J. Comput.*, 29(2): 648–670, 1999.
- [33] O. Shmueli, B. Golany, R. Sayegh, H. Shachnai, M. Perry, N. Gradovitch and B. Yehezkel, “Negotiation Platform”. *International Patent Application* WO 02077759, 2001-2.
- [34] V. V. Vazirani, *Approximation Algorithms*. Springer-Verlag, 2001.
- [35] E. Wolfstetter, “Auctions: An Introduction”. *J. of Economic Surveys*, 10, 1996: 367–420.
- [36] Y. Zhou, “Improved Multi-unit Auction Clearing Algorithms with Interval (Multiple-Choice) Knapsack Problems”. In *Proc. of ISAAC*, 494–506, 2006.