4OR (2023) 21:405–419
https://doi.org/10.1007/s10288-022-00514-4

RESEARCH PAPER

Integer knapsack problems with profit functions of the same value range

Evgeny Gurevsky[1] · Dmitry Kopelevich[2] · Sergey Kovalev[3] · Mikhail Y. Kovalyov[4]

Received: 17 December 2021 / Revised: 17 December 2021 / Accepted: 13 May 2022 /
Published online: 21 June 2022

Abstract

Integer knapsack problems with profit functions of the same value range are studied. Linear time algorithms are presented for the case of convex non-decreasing profit functions, and an NP-hardness proof and a fully polynomial-time approximation scheme are provided for the case of arbitrary non-negative non-decreasing profit functions. Fast solution procedures are also devised for the bottleneck counterparts of these problems. Computational complexity of the case with concave profit functions remains open.

Keywords Knapsack problem · Convex optimization · Median finding · NP-hardness · FPTAS

Mathematics Subject Classification 90C27 · 90C39

1 Introduction

Knapsack problems appear in diverse applications as sub-tasks of more complex problems. Their solution techniques are employed as subroutines in column generation schemes and lower bound constructions for many NP-hard combinatorial optimization problems. Numerous results exist for knapsack problems, see monographs of Martello and Toth (1990) and Kellerer et al. (2004).

Denote an interval of integer numbers $\{a, a+1, \dots, b\}$ as $[a, b]$ for $a < b$, and denote a variable vector $(x_1, \dots, x_n)$ as x. Let integer numbers B, d_j, real number C, and functions $f_j(x_j)$, $j \in [1, n]$, be given such that $d_j > 0$, $0 < B < \sum_{j\in[1,n]} d_j$,

✉ Evgeny Gurevsky
evgeny.gurevsky@univ-nantes.fr

[1] LS2N, Université de Nantes, Nantes, France

[2] Belarusian State University, Minsk, Belarus

[3] INSEEC Business School, Lyon, France

[4] United Institute of Informatics Problems, NASB, Minsk, Belarus

$0 < C \le n$, $f_j(x_j)$ is non-decreasing for $x_j \in [0, d_j]$, $f_j(0) = 0$, $f_j(d_j) = 1$, and $f_j(x_j) = -\infty$ for $x_j \notin [0, d_j]$, $j \in [1, n]$. We study the following two knapsack problems, which we denote as K1 and K2.

Problem K1:

$$\max_x \sum_{j\in[1,n]} f_j(x_j), \text{ subject to}$$

$$\sum_{j\in[1,n]} x_j \le B,$$

$$x_j \in [0, d_j],\ \ j \in [1, n].$$

Problem K2:

$$\min_x \sum_{j\in[1,n]} x_j,\ \text{ subject to}$$

$$\sum_{j\in[1,n]} f_j(x_j) \ge C,$$

$$x_j \in [0, d_j],\ \ j \in [1, n].$$

The problems K1 and K2 admit several equivalent formulations, in which feasible domains of the variables are extended from $[0, d_j]$ to $[a_j, b_j]$, $j \in [1, n]$, the common objective function value range is extended from the interval [0, 1] of real numbers to an arbitrary common interval of real numbers, and the direction of optimization in K1 and the inequality sign in K2 are reversed. Let numbers B^+, C^+, L^+, U^+, B^-, C^-, L^-, U^-, a_j, b_j, and functions $f_j^+(x_j)$ and $f_j^-(x_j)$ be given such that $a_j < b_j$ and

- $f_j^+(a_j) = L^+$, $f_j^+(b_j) = U^+$, $L^+ < U^+$, $f_j^+(x_j)$ is non-decreasing for $x_j \in [a_j, b_j]$, $f_j^+(x_j) = -\infty$ for $x_j \notin [a_j, b_j]$, $j \in [1, n]$,
- $f_j^-(a_j) = U^-$, $f_j^-(b_j) = L^-$, $L^- < U^-$, $f_j^-(x)$ is non-increasing for $x_j \in [a_j, b_j]$, and $f_j^-(x_j) = +\infty$ for $x_j \notin [a_j, b_j]$, $j \in [1, n]$.

The following knapsack problems are equivalent to K1.

Problem K1$^+$:

$$\max_x \sum_{j\in[1,n]} f_j^+(x_j),\ \text{ subject to}$$

$$\sum_{j\in[1,n]} x_j \le B^+,$$

$$x_j \in [a_j, b_j],\ \ j \in [1, n].$$

Problem K1$^-$:

$$\min_x \sum_{j\in[1,n]} f_j^-(x_j),\ \text{ subject to}$$

$$\sum_{j\in[1,n]} x_j \le B^-,$$

$$x_j \in [a_j, b_j],\ \ j \in [1, n].$$

The following problems are equivalent to K2.

Problem K2$^+$:

$$\min_x \sum_{j\in[1,n]} x_j,\ \text{ subject to}$$

$$\sum_{j\in[1,n]} f_j^+(x_j) \ge C^+,$$

$$x_j \in [a_j, b_j],\ \ j \in [1, n].$$

Problem K2$^-$:

$$\min_x \sum_{j\in[1,n]} x_j,\ \text{ subject to}$$

$$\sum_{j\in[1,n]} f_j^-(x_j) \le C^-,$$

$$x_j \in [a_j, b_j],\ \ j \in [1, n].$$

It is easy to see that for any instance of the problem K1$^+$ (resp., K2$^+$), the following instance of K1 (resp., K2) is equivalent: $d_j = b_j - a_j$, $f_j(x_j) = \frac{f_j^+(x_j+a_j)-L^+}{U^+-L^+}$, $j \in [1,n]$, $B = B^+ - \sum_{j\in[1,n]} a_j$ (resp., $C = \frac{C^+-nL^+}{U^+-L^+}$). Similarly, for any instance of the problem K1$^-$ (resp., K2$^-$), the following instance of K1$^+$ (resp., K2$^+$) is equivalent: $f_j^+(x_j) = -f_j^-(x_j)$, $j \in [1,n]$, $B^+ = B^-$ (resp., $C^+ = -C^-$), $L^+ = -U^-$, $U^+ = -L^-$. Furthermore, if the profit functions are convex non-decreasing in the original instance of the problem K1 (resp., K2), then they are concave non-increasing in the corresponding instance of the problem K1$^-$ (resp., K2$^-$).

To facilitate the presentation, it is convenient to assume that all the functions considered in this paper are computable in a constant time. In this case, any instance of any of the problems K1$^+$ and K1$^-$ (resp., K2$^+$ and K2$^-$) can be transformed into an equivalent instance of K1 (resp., K2) in $O(n)$ time.

Motivation for the problem K1$^+$ (resp., K1$^-$) comes from the maximization of the total success (resp., minimization of the total risk) of the projects by allocating limited resources to them. In this situation, there are n projects, $f_j^+(x_j)$, $0 \le f_j^+(x_j) \le 1$, is a convex non-decreasing success function (resp., $f_j^-(x_j)$, $0 \le f_j^-(x_j) \le 1$, is a concave non-increasing risk function) for project j, which depends of the amount x_j of a discrete resource allocated to this project, $x_j \in [a_j, b_j]$. Equality $f_j^+(a_j) = L^+ = 0$ (resp., $f_j^-(a_j) = U^- = 1$) implies that if the resource is allocated at its lower bound, then there is no success (resp., there is a full risk) and equality $f_j^+(b_j) = U^+ = 1$ (resp., $f_j^-(b_j) = L^- = 0$) implies that if the resource is allocated at its upper bound, then there is a full success (resp., there is no risk).

Problems K1 and K2 belong to the class of non-linear knapsack problems, which consist in the optimization of a non-linear function, subject to a single non-linear constraint. A review of these problems and their solution methods is given by Bretthauer and Shetty (2002). More recent results can be found in D'Ambrosio and Martello (2011), Zhang and Chen (2012) and D'Ambrosio et al. (2018). In these studies, portfolio selection, stratified sampling, production and inventory planning, and resource distribution are noted as applications of the non-linear knapsack problems.

Given a function $f(x)$ of a discrete argument $x \in [a, b]$ and a number A, we call function $\min\{x \in [a, b] \mid f(x) \ge A\}$ a *minimum pseudo-inverse* of $f(x)$. In Sect. 2, the case of convex profit functions is studied. We present $O(n)$ time algorithms for the problems K1 and K2 under the assumptions that the function $f_j(x_j)$ for any x_j in the problem K1 and its minimum pseudo-inverse $\min\{x_j \in [0, d_j] \mid f_j(x_j) \ge A\}$ for any A in the problem K2 are computable in a constant time for any $j \in [1, n]$. These algorithms are adaptations of the algorithms for the problems K1$^-$ and K2$^-$ with concave non-increasing functions $f_j^-(x_j)$, $j \in [1, n]$, given by Gurevsky et al. (2022) as an illustration of their generic solution approach for a class of min-sum controllable risk problems. The algorithms employ the median finding technique of Blum et al. (1973), which was used for the fractional knapsack problem by Balas and Zemel (1980).

The case of arbitrary non-negative non-decreasing profit functions is considered in Sect. 3. Let a number $\varepsilon > 0$ be given. An algorithm for an optimization problem is called an *ε-approximation algorithm*, if for any instance I of this problem it finds a feasible solution with value $F^0(I)$ such that $|F^0(I) - F^*(I)| \leq \varepsilon|F^*(I)|$, where $F^*(I)$ is the optimum value for the instance I. A *Fully Polynomial-Time Approximation Scheme (FPTAS)* for an optimization problem is a family of ε-approximation algorithms $\{A_\varepsilon\}$ each of which runs in time bounded by a polynomial of $1/\varepsilon$ and of the problem input length in binary encoding. FPTASes are popular solution techniques for knapsack type problems. Recently, Halman et al. (2018) presented an FPTAS for a knapsack problem with parametric weights and Malaguti et al. (2019) and Kovalev (2021) provided FPTASes for a fractional knapsack problem. In Sect. 3, we prove that the problems K1 and K2 with non-negative non-decreasing profit functions are ordinary NP-hard, and present FPTASes with the running times $O(n^3 \log_2 n + \frac{n^3}{\varepsilon^2})$ and $O(n^3 \log_2(nd_{\max}) + \frac{n^3}{\varepsilon^2})$, respectively, where $d_{\max} = \max_{j\in[1,n]} d_j$. Our FPTASes are applications of the trimming-of-the-state-space and rounding techniques, originally described by Ibarra and Kim (1975) and Sahni (1977) for the classical knapsack and subset sum problems. We did not manage to prove NP-hardness or develop a polynomial-time algorithm for the case of concave profit functions. Bottleneck counterparts of the problems K1 and K2 are formulated in Sect. 3.3. Simple solution procedures are presented for them. The paper concludes with a short summary of the results and suggestions for future research.

2 Convex profit functions

In this section, it is assumed that the profit functions $f_j(x_j)$, $j \in [1,n]$, are convex non-decreasing. In this case, $O(n)$ time algorithms are developed for the problems K1 and K2.

2.1 Problem K1

Algorithm for the problem K1 is based on the following theorems.

Theorem 1 *There exists an optimal solution x of the problem* K1*, in which there is at most one index $j^* \in [1,n]$ such that all variables but variable x_{j^*} are equal to their lower or upper bounds: $x_j \in \{0, d_j\}$, $j \in [1,n]\backslash\{j^*\}$, and $x_{j^*} = \min\left\{d_{j^*}, B - \sum_{j\in[1,n]\backslash\{j^*\}} x_j\right\}$.*

Proof Follows from the convexity of the functions $f_j(x_j)$, $j \in [1,n]$. □

Denote $N_j = \{i \mid d_i \leq d_j, i \in [1,n]\}$, $j \in [1,n]$.

Theorem 2 *There exists an optimal solution x of the problem* K1 *and indices j^0 and j^*, $j^0 \in [1,n]$, $j^* \in [1,n]$, such that $x_j = d_j$ for $j \in N_{j^0}\backslash\{j^*\}$, $x_j = 0$ for $j \notin N_{j^0} \cup \{j^*\}$ and $x_{j^*} = \min\left\{d_{j^*}, B - \sum_{j\in N_{j^0}\backslash\{j^*\}} d_j\right\}$.*

Proof Let index j^*, mentioned in Theorem 1, be fixed. Since $f_j(d_j) = 1 \geq f_i(x_i)$ for $j \in [1,n]$, $i \in [1,n]$, the problem K1 reduces to selecting the maximum number of d_j, $j \neq j^*$, whose sum does not exceed B. Obviously, these numbers form the set $N_{j^0} \backslash \{j^*\}$ for some $j^0 \in [1, n]$. □

Denote $D_j = \sum_{i \in N_j} d_i$, $j \in [1, n]$.

Theorem 3 *If the function $f_j(x_j)$ is computable in a constant time for any x_j and $j \in [1, n]$, then the problem* K1 *can be solved in $O(n)$ time.*

Proof Assume for a while that the elements of the set $[1, n]$ are re-numbered such that $d_1 \leq \cdots \leq d_n$. Define a unique index $j^{(B)}$ such that $D_{j^{(B)}} \leq B < D_{j^{(B)}+1}$, $j^{(B)} \in [0, n-1]$, where $D_0 := 0$. Denote by F^* the optimal solution value of the problem K1 and denote by $F(v, u)$ the value of a solution x of this problem such that $x_j = d_j$ for $j \in N_v \backslash \{u\}$, $x_j = 0$ for $j \notin (N_v \cup \{u\})$ and $x_u = \min \left\{ d_u, B - \sum_{j \in N_v \backslash \{u\}} d_j \right\}$, that is,

$$F(v, u) = \sum_{j \in N_v \backslash \{u\}} f_j(d_j) + f_u(x_u) = |N_v \backslash \{u\}| + f_u \left(\min \left\{ d_u, B - \sum_{j \in N_v \backslash \{u\}} d_j \right\} \right).$$

Note that, by the definition of the function $f_j(x_j)$, $F(v, u) = -\infty$ if $x_u < 0$. It follows from Theorem 2 that $F^* = \max\{F(v, u) \mid v \in [1, n], u \in [1, n]\}$.

Consider three cases for the index u: 1) $u \leq j^{(B)}$, 2) $u = j^{(B)} + 1$ and 3) $u \geq j^{(B)} + 2$. In the case 1), there are three sub-cases: 1.1) $v \leq j^{(B)}$, 1.2) $v = j^{(B)} + 1$, and 1.3) $v \geq j^{(B)} + 2$. In the sub-case 1.1), $F(v, u) = v \leq j^{(B)}$. In the sub-case 1.2), there are two more sub-cases: 1.2.1) $\sum_{j \in N_{j^{(B)}+1} \backslash \{u\}} d_j \leq B$ and 1.2.2) $\sum_{j \in N_{j^{(B)}+1} \backslash \{u\}} d_j > B$. In the sub-case 1.2.1), $F(v, u) = F(j^{(B)} + 1, u) = j^{(B)} + f_u(B - D_{j^{(B)}+1} + d_u)$, and in the sub-case 1.2.2), $F(v, u) = F(j^{(B)} + 1, u) = -\infty$. In the case 2), there are three sub-cases again: 2.1) $v \leq j^{(B)}$, 2.2) $v = j^{(B)} + 1$, and 2.3) $v \geq j^{(B)} + 2$. In the sub-case 2.1), $F(v, u) = F(v, j^{(B)} + 1) = v + f_{j^{(B)}+1}(\min\{d_{j^{(B)}+1}, B - D_v\}) \leq j^{(B)}$ if $v \leq j^{(B)} - 1$, and $F(v, u) = F(j^{(B)}, j^{(B)} + 1) = j^{(B)} + f_{j^{(B)}+1}(B - D_{j^{(B)}})$ if $v = j^{(B)}$. In the sub-case 2.2), $F(v, u) = F(j^{(B)} + 1, j^{(B)} + 1) = j^{(B)} + f_{j^{(B)}+1}(B - D_{j^{(B)}})$, and in the sub-case 2.3), $F(v, u) = -\infty$. In the case 3), there are two sub-cases: 3.1) $v \leq j^{(B)}$, and 3.2) $v \geq j^{(B)} + 1$. In the sub-case 3.1), $F(v, u) = v + f_u(\min\{d_u, B - D_v\}) \leq j^{(B)}$ if $v \leq j^{(B)} - 1$, and $F(v, u) = F(j^{(B)}, u) = j^{(B)} + f_u(B - D_{j^{(B)}})$ if $v = j^{(B)}$. In the sub-case 3.2), $F(v, u) = -\infty$. We deduce that

$$F^* = j^{(B)} + \max \Big\{ \max\{f_u(B - D_{j^{(B)}+1} + d_u) \mid u \in N_{j^{(B)}}\},$$
$$\max\{f_u(B - D_{j^{(B)}}) \mid u \in [1, n] \backslash N_{j^{(B)}}\} \Big\}.$$

If index of the $j^{(B)}$-th smallest value d_j is given, then the sets $N_{j^{(B)}}$, $N_{j^{(B)}+1}$ and $[1, n] \backslash N_{j^{(B)}}$, and the values $D_{j^{(B)}}$ and $D_{j^{(B)}+1}$ can be calculated in $O(n)$ time. Hence, the optimal value F^* and the corresponding optimal vector x^* can be found in $O(n)$ time as well. The index $j^{(B)}$ is an analogue of the *split item* in the continuous knapsack problem, which can be found in $O(n)$ time by the algorithm SPLIT described in (Kellerer et al. 2004, p. 44). □

Observe that the algorithm in the proof of Theorem 3 is an adaptation of the greedy algorithm of Dantzig (1957) for the continuous knapsack problem in a combination with the median finding technique. However, we do not see a way to justify this observation other than to repeat our arguments in Theorems 1, 2 and 3.

2.2 Problem K2

Similar to K1, an $O(n)$ time algorithm for the problem K2 is justified by the following theorems.

Theorem 4 *There exists an optimal solution x of the problem* K2*, in which there is at most one index $j^* \in [1, n]$ such that all variables but variable x_{j^*} are equal to their lower or upper bounds: $x_j \in \{0, d_j\}$, $j \in [1, n] \backslash \{j^*\}$, and $x_{j^*} = \min \Big\{ x \mid x \in [0, d_{j^*}],\ f_{j^*}(x) \geq C - \sum_{j \in [1,n] \backslash \{j^*\}} f_j(x_j) \Big\}$.*

Proof Follows from the convexity of the functions $f_j(x_j)$, $j \in [1, n]$. □

Recall that $N_j = \{i \mid d_i \leq d_j, i \in [1, n]\}$ and $D_j = \sum_{i \in N_j} d_i$, $j \in [1, n]$.

Theorem 5 *There exists an optimal solution x of the problem* K2 *and indices j^0 and j^*, $j^0 \in [1, n]$, $j^* \in [1, n]$, such that $x_j = d_j$ for $j \in N_{j^0} \backslash \{j^*\}$, $x_j = 0$ for $j \notin (N_{j^0} \cup \{j^*\})$ and $x_{j^*} = \min \Big\{ x \mid x \in [0, d_{j^*}],\ f_{j^*}(x) \geq C - |N_{j^0} \backslash \{j^*\}| \Big\}$.*

Proof Let the index j^* mentioned in Theorem 4 be fixed. Since $f_j(d_j) = 1 \geq f_i(x_i)$ for $j \in [1, n]$, $i \in [1, n]$, the problem K2 reduces to selecting the minimum number, denoted as k, of smallest d_j, $j \neq j^*$, such that $k \geq C - 1$. Obviously, these numbers form the set $N_{j^0} \backslash \{j^*\}$ for some $j^0 \in [1, n]$. □

Theorem 6 *If the minimum pseudo-inverse* $\min\{x_j \in [0, d_j] \mid f_j(x_j) \geq A\}$ *of the function $f_j(x_j)$ is computable in a constant time for any A and $j \in [1, n]$, then the problem* K2 *can be solved in $O(n)$ time.*

Proof Assume for a while that $d_1 \leq \cdots \leq d_n$. Denote by G^* the optimal solution value of the problem K2 and denote by $G(v, u)$ the value of a solution x of this problem such that $x_j = d_j$ for $j \in N_v \backslash \{u\}$, $x_j = 0$ for $j \notin (N_v \cup \{u\})$ and $x_u = \min \Big\{ x \mid x \in [0, d_u],\ f_u(x) \geq C - |N_v \backslash \{u\}| \Big\}$, that is,

$$G(v, u) = \sum_{j \in N_v \backslash \{u\}} d_j + x_u = \sum_{j \in N_v \backslash \{u\}} d_j + \min\{x \mid x \in [0, d_u],\ f_u(x) \geq C - |N_v \backslash \{u\}|\}.$$

If the system of two relations $x \in [0, d_u]$ and $f_u(x) \geq C - |N_v \backslash \{u\}|$ has a solution (that is, if $C - |N_v \backslash \{u\}| \leq 1$), then $G(v, u)$ is well defined. If it does not have a solution (that is, if $C - |N_v \backslash \{u\}| > 1$), then define $G(v, u) = +\infty$. It follows from Theorem 5, that $G^* = \min\{G(v, u) \mid v \in [1, n], u \in [1, n], v \geq \lceil C \rceil - 1\}$.

Consider $v \geq \lceil C \rceil - 1$ and two cases for the index u: 1) $1 \leq u \leq v$ and 2) $u \geq v + 1$. In the case 1), $|N_v \backslash \{u\}| = v - 1$ and there are three sub-cases: 1.1) $v = \lceil C \rceil - 1$, 1.2) $v = \lceil C \rceil$, and 1.3) $v \geq \lceil C \rceil + 1$. In the sub-case 1.1), $|N_v \backslash \{u\}| = \lceil C \rceil - 2$, $C - |N_v \backslash \{u\}| = C - \lceil C \rceil + 2 > 1$, and hence, $G(v, u) = +\infty$. In the sub-case 1.2), $C - |N_v \backslash \{u\}| = C - v + 1 = C - \lceil C \rceil + 1 \leq 1$, and hence, for $u \leq v = \lceil C \rceil$,

$$G(v, u) = G(\lceil C \rceil, u) = D_{\lceil C \rceil} - d_u + \min\{x \mid x \in [0, d_u],\ f_u(x) \geq C - \lceil C \rceil + 1\}.$$

In the sub-case 1.3), $C - |N_v \backslash \{u\}| = C - v + 1 \leq C - \lceil C \rceil \leq 0$, and hence,

$$\begin{aligned} G(v, u) &= D_v - d_u + \min\{x \mid x \in [0, d_u],\ f_u(x) \geq C - v + 1\} \\ &= D_v - d_u \geq D_v - d_v = D_{v-1} \geq D_{\lceil C \rceil}. \end{aligned}$$

In the case 2), $|N_v \backslash \{u\}| = v$ and there are two sub-cases: 2.1) $v = \lceil C \rceil - 1$ and 2.2): $v \geq \lceil C \rceil$. In the sub-case 2.1), $|N_v \backslash \{u\}| = v = \lceil C \rceil - 1$, $C - |N_v \backslash \{u\}| = C - \lceil C \rceil + 1 \leq 1$, and hence, for $u \geq v + 1 = \lceil C \rceil$,

$$G(v, u) = G(\lceil C \rceil - 1, u) = D_{\lceil C \rceil - 1} + \min\{x \mid x \in [0, d_u],\ f_u(x) \geq C - \lceil C \rceil + 1\}.$$

In the sub-case 2.2), $C - |N_v \backslash \{u\}| = C - v \leq C - \lceil C \rceil \leq 0$, and hence,

$$G(v, u) = D_v + \min\{x \mid x \in [0, d_u],\ f_u(x) \geq C - v\} = D_v \geq D_{\lceil C \rceil}.$$

We deduce that

$$\begin{aligned} G^* = \min \Bigg\{ & D_{\lceil C \rceil} - \max_{1 \leq u \leq \lceil C \rceil} \{d_u - \min\{x \mid x \in [0, d_u],\ f_u(x) \geq C - \lceil C \rceil + 1\}\}, \\ & D_{\lceil C \rceil - 1} + \min_{\lceil C \rceil \leq u \leq n} \min\{x \mid x \in [0, d_u],\ f_u(x) \geq C - \lceil C \rceil + 1\} \Bigg\}. \end{aligned}$$

Similar to the index $j^{(B)}$ in the Proof of Theorem 3, index of the $\lceil C \rceil$-th smallest value d_j, and hence, the optimal value G^* and the corresponding optimal solution of the problem K2 can be found in $O(n)$ time by an adaptation of the algorithm SPLIT in (Kellerer et al. 2004, p. 44). □

Note that the minimum pseudo-inverse is computable in a constant time for many functions. For example, for the function $f(x) = x^2$, we have $\min\{x \in [0, d] \mid f(x) \geq A\} = \lceil \sqrt{A} \rceil$, if $\sqrt{A} \leq d$ and the above minimum does not exist if $\sqrt{A} > d$. For an arbitrary non-decreasing function $f(x)$ which is computable in a constant time, and $x \in [0, d]$, the minimum pseudo-inverse can be calculated in $O(\log_2 d)$ time by a bisection search.

3 Arbitrary non-negative non-decreasing profit functions

In this section, we assume that the profit functions $f_j(x_j)$, $j \in [1,n]$, are arbitrary non-negative non-decreasing and their minimum pseudo-inverse $\min\{x_j \in [0,d_j] \mid f_j(x_j) \geq A\}$ is computable in a constant time for any A and $j \in [1,n]$. We prove that any of the problems K1 and K2 is NP-hard in this case and present FPTASes for K1 and K2. We failed to prove NP-hardness or develop a polynomial-time algorithm for the problem K1 or K2 with concave profit functions. Note that analogues of Theorems 1 and 4 do not hold in this case.

3.1 NP-hardness proof

Let us consider the following decision version of any of the problems K1 and K2, which we denote as KD.

Problem KD*: Given positive integer numbers A, B, d_j and arbitrary non-negative non-decreasing functions $f_j(\cdot)$, $j \in [1,n]$, does there exist a vector $x = (x_1,\ldots,x_n)$ such that $\sum_{j\in[1,n]} f_j(x_j) \geq A$, $\sum_{j\in[1,n]} x_j \leq B$ and $x_j \in [0,d_j]$, $j \in [1,n]$?*

We prove that KD *is NP-complete if profit functions $f_j(\cdot)$ are arbitrary non-negative non-decreasing and computable in polynomial time of $\log_2 d_j$, $j \in [1,n]$. This implies that both* K1 *and* K2 *are NP-hard.*

Theorem 7 *Problem* KD *is NP-complete if profit functions $f_j(\cdot)$ are arbitrary non-negative non-decreasing and computable in polynomial time of $\log_2 d_j$, $j \in [1,n]$.*

Proof We use a transformation from the NP-complete problem Partition (Garey and Johnson 1979): Given positive integer numbers h_j, $j \in [1,k]$, and H such that $\sum_{j\in[1,k]} h_j = 2H$, is there a set $I \subset [1,k]$ such that $\sum_{j\in I} h_j = H$? Assume without loss of generality that $h_j \geq 2$, $j \in [1,k]$.

For any instance of Partition, we construct the following instance of KD. Set $n = k$, $A = 2$, $B = H$, $d_j = H-1$, $f_j(x_j) = 0$ for $x_j \in [0, h_j - 1]$, $f_j(x_j) = 2h_j/H$ for $x_j \in [h_j, d_j - 1]$ and $f_j(d_j) = 1$, $j \in [1,n]$. We show that the instance of Partition has a solution if and only if the respective instance of KD has a solution.

Assume that set I is a solution of the instance of Partition. Define $x_j = h_j$ if $j \in I$ and $x_j = 0$ if $j \notin I$, $j \in [1,n]$. Then, we have $\sum_{j\in[1,n]} f_j(x_j) = 2\sum_{j\in I} h_j/H = 2$ and $\sum_{j\in I} x_j = \sum_{j\in I} h_j = H$. Hence, vector x is a solution of the instance of KD.

Now, assume that the instance of KD has a solution, which we denote as vector x. If there is $j \in [1,n]$ such that $x_j = d_j = H-1$, then $f_j(x_j) = 1$, $\sum_{i\in[1,j-1]\cup[j+1,n]} x_i \leq 1$, $x_i \leq 1$, $i \in [1,j-1] \cup [j+1,n]$, and, as a consequence, $\sum_{j\in[1,n]} f_j(x_j) = 1$, which is a contradiction. Therefore, $x_j \leq d_j - 1$, $j \in [1,n]$. Define set I such that $j \in I$ if $d_j - 1 \geq x_j \geq h_j$ and $j \notin I$ if $x_j < h_j$, $j \in [1,n]$. Then, we have $f_j(x_j) = 0$ for $j \notin I$ and $f_j(x_j) = f_j(h_j) = 2h_j/H$ for $j \in I$. Taking into account the definition of the problem KD, we must have $\sum_{j\in I} h_j \leq \sum_{j\in I} x_j \leq \sum_{j\in[1,n]} x_j \leq H$ and $\sum_{j\in I} 2h_j/H = \sum_{j\in I} f_j(h_j) = \sum_{j\in I} f_j(x_j) + \sum_{j\notin I} f_j(x_j) = \sum_{j\in[1,n]} f_j(x_j) \geq 2$. The latter chain of relations implies $\sum_{j\in I} h_j \geq H$, which together with $\sum_{j\in I} h_j \leq H$ further indicates that I is a solution of the instance of Partition. □

3.2 Generic FPTASes

The main components of our FPTAS for the problems K1 are:

- formulation of a generic problem, denoted as KG1,
- formulation of a scaled and rounded problem, denoted as KG-R1,
- proof that an optimal solution of KG-R1 is an ε-approximation solution of KG-R1,
- dynamic programming algorithm for KG-R1, denoted as DP-R1, which constitutes an FPTAS for KG1, and
- a procedure to improve lower and upper bounds for the optimal profit F^*, which finds a value L such that $L \le F^* \le 4L$ and makes the FPTAS more efficient.

Assume that, besides $f_j(x_j)$, non-negative non-decreasing functions $g_j(x_j)$ are given, $j \in [1, n]$.

Generic problem KG1:

$$\max_x \sum_{j \in [1,n]} f_j(x_j), \text{ subject to } \sum_{j \in [1,n]} g_j(x_j) \le B, \ x_j \in [0, d_j], \ j \in [1, n].$$

Let x^* and F^* denote an optimal solution and its optimal objective value of KG1, respectively. Assume that numbers V and U are known such that $0 < V \le F^* \le U$. Define $\delta = \varepsilon V/n$.

Scaled and rounded problem KG-R1:

$$\max_x \sum_{j \in [1,n]} \left\lceil \frac{f_j(x_j)}{\delta} \right\rceil, \text{ subject to } \sum_{j \in [1,n]} g_j(x_j) \le B, \ x_j \in R_j, \ j \in [1, n],$$

where

$$\begin{aligned} R_j &= \left\{ r_j(0), r_j(1), \ldots, r_j\left(\left\lfloor \frac{U}{\delta} \right\rfloor\right) \right\}, \\ r_j(s) &= \min \left\{ x \mid x \in [0, d_j], \ \left\lfloor \frac{f_j(x)}{\delta} \right\rfloor = s \right\} \\ &= \min\{x \mid x \in [0, d_j], \ f_j(x) \ge s\delta\}, \ j \in [1, n]. \end{aligned}$$

Let $x^{(0)}$, $Q^{(0)}$ and $F^{(0)}$ denote an optimal solution, its objective function value with respect to the problem KG-R1 and its objective function value with respect to the original generic problem KG1, respectively. Note that $x^{(0)}$ is feasible for KG1, and x^* is feasible for KG-R1. Let us evaluate the quality of $x^{(0)}$ with respect to KG1.

$$\begin{aligned} F^{(0)} &= \sum_{j \in [1,n]} f_j(x_j^{(0)}) = \delta \sum_{j \in [1,n]} \frac{f_j(x_j^{(0)})}{\delta} > \delta \sum_{j \in [1,n]} \left(\left\lceil \frac{f_j(x_j^{(0)})}{\delta} \right\rceil - 1 \right) \\ &\ge [\, x^{(0)} \text{ is optimal for KG-R1}] \\ &\ge \delta \sum_{j \in [1,n]} \left(\left\lceil \frac{f_j(x_j^*)}{\delta} \right\rceil - 1 \right) \ge \sum_{j \in [1,n]} f_j(x_j^*) - n\delta = F^* - n\delta = F^* - \varepsilon V. \end{aligned}$$

We obtain $F^* - F^{(0)} < \varepsilon F^*$. Let us evaluate $Q^{(0)}$ from above.

$$\begin{aligned} Q^{(0)} &= \sum_{j\in[1,n]} \left\lceil \frac{f_j(x^{(0)})}{\delta} \right\rceil < \sum_{j\in[1,n]} \left(\frac{f_j(x^{(0)})}{\delta} + 1 \right) \\ &\le \sum_{j\in[1,n]} \frac{f_j(x^*)}{\delta} + n = \frac{F^*}{\delta} + n \le \frac{U}{\delta} + n \\ &= \frac{nU}{\varepsilon V} + n. \end{aligned}$$

We deduce that $Q^{(0)} \le \left\lfloor \frac{nU}{\varepsilon V} \right\rfloor + n := U^{(0)}$.

In the algorithm DP-R1 for the problem KG-R1, partial solutions $x = (x_1, \ldots, x_j)$ are constructed for $j \in [1, n]$. With each partial solution x, *state* (j, Q) is associated, where $Q = \sum_{h\in[1,j]} \left\lceil \frac{f_h(x_h)}{\delta} \right\rceil$ is the current value of the *scaled and rounded profit*. Constraint function $G(j, Q)$ is recursively calculated for each state (j, Q), which is the minimum value of $\sum_{h\in[1,j]} g_h(x_h)$ over partial solutions x in the same state (j, Q). A partial solution with value $G(j, Q)$ *dominates* all other partial solutions in the state (j, Q) in the sense that if there is a partial solution in this state that can be extended to an optimal solution of KG-R1, then the dominant solution can also be extended to an optimal solution of KG-R1. Therefore, only the dominant solution can be left for each state, and non-dominant solutions can be eliminated.

Algorithm DP-R1. The initialization is $G(0, 0) = 0$, and $G(0, Q) = B + 1$ for $(j, Q) \ne (0, 0)$, $j \in [1, n]$, $Q = 0, 1, \ldots, U^{(0)}$. The recursion for $j \in [1, n]$, $Q = 0, 1, \ldots, U^{(0)}$, is

$$G(j, Q) = \min_{x_j \in R_j} \left\{ G\left(j - 1, Q - \left\lceil \frac{f_j(x_j)}{\delta} \right\rceil \right) + g_j(x_j) \right\}.$$

The optimal objective function value of the problem KG-R1 is equal to

$$Q^{(0)} = \max\{Q \mid G(n, Q) \le B, Q = 0, 1, \ldots, U^{(0)}\},$$

and the corresponding optimal solution can be found by tracing back optimal solutions of the recursive equation. The recursive equation can be solved in $O(|R_j|) = O\left(\frac{nU}{\varepsilon V}\right)$ time for each state (j, Q). Therefore, the running time of the algorithm DP-R1 is $O\left(\frac{n^3U^2}{\varepsilon^2V^2}\right)$.

Kovalev (2021) describes a procedure to improve lower and upper bounds for a minimization problem if an approximation algorithm satisfying certain properties is available. It is given in the Appendix for completeness. The bound improvement procedure with algorithm DP-R1 incorporated in it runs in $O(n^3 \log_2 \frac{U}{V})$ time and delivers number L such that $L \le F^* \le 4L$ for the problem KG1. With lower and upper bounds $V := L$ and $U := 4L$, algorithm DP-R1 runs in $O\left(\frac{n^3}{\varepsilon^2}\right)$ time. Therefore, the overall running time of the FPTAS for KG1 is $O\left(n^3 \log_2 \frac{U}{V} + \frac{n^3}{\varepsilon^2}\right)$. For KG1, the lower bound can be calculated as $V = \max_{j\in[1,n]}\{f_j(x_j^0)\}$, where $x_j^0 = \min\{d_j, \max\{x_j \mid g_j(x_j) \le B\}\}$, $j \in [1,n]$. In this case, the corresponding upper bound is $U = nV$, and the running time of the FPTAS for KG1, and hence, for K1, is $O(n^3 \log_2 n + \frac{n^3}{\varepsilon^2})$.

Similar ideas – a generic problem KG2, a scaled and rounded problem KG-R2, a dynamic programming algorithm for KG-R2, which constitutes an FPTAS for KG2, and a bound improvement procedure – are used to develop an FPTAS for the minimization problem K2.

Generic problem KG2:

$$\min_x \sum_{j\in[1,n]} g_j(x_j), \text{ subject to } \sum_{j\in[1,n]} f_j(x_j) \ge C,\ x_j \in [0,d_j],\ j \in [1,n].$$

Let x^* now denote an optimal solution of the problem KG2, and let G^* denote its optimal objective function value. Assume that numbers S and T are known such that $0 < S \le G^* \le T$. Define $\delta = \varepsilon S/n$.

Scaled and rounded problem KG-R2:

$$\min_x \sum_{j\in[1,n]} \left\lfloor \frac{g_j(x_j)}{\delta} \right\rfloor, \text{ subject to } \sum_{j\in[1,n]} f_j(x_j) \ge C,\ x_j \in H_j,\ j \in [1,n],$$

where

$$\begin{aligned} H_j &= \left\{h_j(0), h_j(1), \ldots, h_j\left(\left\lfloor \frac{T}{\delta} \right\rfloor\right)\right\},\\ h_j(s) &= \max\left\{x \mid x \in [0,d_j],\ \left\lfloor \frac{g_j(x)}{\delta} \right\rfloor = s\right\}\\ &= \max\{x \mid x \in [0,d_j],\ g_j(x) < (s+1)\delta\},\ j \in [1,n]. \end{aligned}$$

Assume that $h_j(s)$ is calculated in a constant time for any s. In the problem K2, $g_j(x) = x$ and $h_j(s) = \min\{d_j, \lceil (s+1)\delta \rceil - 1\}$. Let $x^{(0)}$, $P^{(0)}$ and $G^{(0)}$ denote an optimal solution, its objective function value with respect to the problem KG-R2 and its objective function value with respect to the problem KG2, respectively. We have

$$\begin{aligned} G^{(0)} &= \sum_{j\in[1,n]} g_j(x_j^{(0)}) = \delta \sum_{j\in[1,n]} \frac{g_j(x_j^{(0)})}{\delta} < \delta \sum_{j\in[1,n]} \left(\left\lfloor \frac{g_j(x_j^{(0)})}{\delta} \right\rfloor + 1\right)\\ &\le [x^{(0)} \text{is optimal for } KG\text{-}R2]\\ &\le \delta \sum_{j\in[1,n]} \left(\left\lfloor \frac{g_j(x_j^*)}{\delta} \right\rfloor + 1\right) \le \sum_{j\in[1,n]} g_j(x_j^*) + n\delta = G^* + n\delta = G^* + \varepsilon S. \end{aligned}$$

Thus, $G^{(0)} - G^* < \varepsilon G^*$. Let us evaluate $P^{(0)}$ from above.

$$P^{(0)} = \sum_{j\in[1,n]} \left\lfloor \frac{g_j(x^{(0)})}{\delta} \right\rfloor \le \sum_{j\in[1,n]} \left\lfloor \frac{g_j(x^*)}{\delta} \right\rfloor \le \sum_{j\in[1,n]} \frac{g_j(x^*)}{\delta} = \frac{G^*}{\delta} \le \frac{T}{\delta} = \frac{nT}{\varepsilon S}.$$

We deduce that $P^{(0)} \leq \left\lfloor \frac{nT}{\varepsilon S} \right\rfloor := T^{(0)}$.

In the algorithm DP-R2 for the problem KG-R2, partial solutions $x = (x_1, \ldots, x_j)$ are constructed for $j \in [1, n]$. With each partial solution x, a *state* (j, P) is associated, where $P = \sum_{h \in [1,j]} \left\lfloor \frac{g_h(x_h)}{\delta} \right\rfloor$ is the current value of the *scaled and rounded cost*. Dominance function $F(j, P)$ is recursively calculated for each state (j, P), which is the maximum value of $\sum_{h \in [1,j]} f_h(x_h)$ over partial solutions x in the same state (j, P).

Algorithm DP-R2. The initialization is $F(0, 0) = 0$, and $F(0, P) = -\sum_{j \in [1,n]} \left\lfloor \frac{g_j(d_j)}{\delta} \right\rfloor$ for $(j, P) \neq (0, 0)$, $j \in [1, n]$, $P = 0, 1, \ldots, T^{(0)}$. The recursion for $j \in [1, n]$, $P = 0, 1, \ldots, T^{(0)}$, is

$$F(j, P) = \max_{x_j \in H_j} \left\{ F\left(j - 1, P - \left\lfloor \frac{g_j(x_j)}{\delta} \right\rfloor\right) + f_j(x_j) \right\}.$$

The optimal objective function value of the problem KG-R2 is equal to

$$P^{(0)} = \min\{P \mid F(n, P) \geq C, P = 0, 1, \ldots, T^{(0)}\},$$

and the corresponding optimal solution can be found by tracing back optimal solutions of the recursive equation. The running time of the algorithm DP-R2 is $O\left(\frac{n^3 T^2}{\varepsilon^2 S^2}\right)$.

Kovalyov (1995) describes a procedure to improve lower and upper bounds for a minimization problem if an approximation algorithm satisfying certain properties is available. The bound improvement procedure with algorithm DP-R2 incorporated in it runs in $O(n^3 \log_2 \frac{T}{S})$ time and delivers number L such that $L \leq F^* \leq 3L$ for the problem KG2. With lower and upper bounds $S := L$ and $T := 3L$, algorithm DP-R2 runs in $O(\frac{n^3}{\varepsilon^2})$ time. Therefore, the overall running time of the ε-approximation algorithm for KG2 is $O(n^3 \log_2 \frac{T}{S} + \frac{n^3}{\varepsilon^2})$.

For KG2, the lower and upper bounds can be calculated as $S = g_{\min}$ and $T = n g_{\max}$, where $g_{\min} = \min_{j \in [1,n]}\{g_j(1)\}$ and $g_{\max} = \max_{j \in [1,n]}\{g_j(d_j)\}$. In this case, the running time of the ε-approximation algorithm for KG2 is $O\left(n^3 \log_2 \frac{n g_{\max}}{g_{\min}} + \frac{n^3}{\varepsilon^2}\right)$. For K2, it becomes $O(n^3 \log_2(n d_{\max}) + \frac{n^3}{\varepsilon^2})$, which means that this algorithm constitutes an FPTAS for K2.

Remark 1 The problem K1 can be solved in $O(nB^2)$ time by a dynamic programming algorithm similar to DP-R1, in which the roles of the dominance function and the state variable are switched. Namely, value $b := \sum_{h \in [1,j]} x_h \in [0, B]$ is the state variable and $F(j, b) := \sum_{h \in [1,j]} f_h(x_h)$ is the dominance function to be maximized. The problem K2 can be solved in $O((\sum_{j \in [1,n]} d_j)^2)$ time by the algorithm DP-R2, in which $g_j(x_j) = x_j$, $j \in [1, n]$, and $\delta = 1$. Thus, both problems K1 and K2 are pseudo-polynomially solvable.

3.3 Bottleneck problems

The following bottleneck counterparts of the problems K1 and K2 can be of interest. Let E be a given number, $0 \leq E \leq 1$.

Problem K1B:

$$\max_{x} \min_{j \in [1,n]} f_j(x_j), \text{ subject to}$$

$$\sum_{j \in [1,n]} x_j \leq B,$$

$$x_j \in [0, d_j], \; j \in [1, n].$$

Problem K2B:

$$\min_{x} \sum_{j \in [1,n]} x_j, \text{ subject to}$$

$$\min_{j \in [1,n]} f_j(x_j) \geq E,$$

$$x_j \in [0, d_j], \; j \in [1, n].$$

Consider the problem K2B and define $f_j^{(-1)}(E) = \min\{x_j \mid f_j(x_j) \geq E, x_j \in [0, d_j]\}, \; j \in [1, n]$. It is obvious that the vector x^* in which $x_j^* = f_j^{(-1)}(E), j \in [1, n]$, is optimal for K2B. Therefore, this problem can be solved in $O(n)$ time, provided that the value $f_j^{(-1)}(E)$ is computable in a constant time for any $j \in [1, n]$.

Consider the problem K1B. Assume that the difference between any two distinct values $f_j(x_j)$, $j = 1, \ldots, n$, is at least $\frac{1}{\delta}$ for a positive integer δ, and that the value $f_j^{(-1)}(A)$ is computable in a constant time for any $j \in [1, n]$ and A. In this case, the problem K1B can be solved in $O(n \log_2 \delta)$ time by a bisection search in the interval [0, 1] of the optimal objective function value F^* of this problem. In each iteration of this search, the relation $\sum_{j \in [1,n]} f_j^{(-1)}(A) \leq B$ is verified for a trial value A, $LB \leq A \leq UB$, starting with lower bound $LB = 0$ and upper bound $UB = 1$. If the mentioned relation is satisfied, then $F^* \geq A$, else $F^* < A$. The lower and upper bounds LB and UB are re-set accordingly. The procedure stops when $UB - LB < \frac{1}{\delta}$, in which case the vector x^* with $x_j^* = f_j^{(-1)}(LB)$, $j \in [1, n]$, is optimal for K1B.

4 Conclusions and suggestions for future research

The following results are obtained:

- an $O(n)$ time algorithm for the problem K1, provided that the profit function $f_j(x_j)$ is convex non-decreasing and computable in a constant time for any $x_j \in [0, d_j]$ and $j \in [1, n]$,
- an $O(n)$ time algorithm for the problem K2, provided that the profit function $f_j(x_j)$ is convex non-decreasing and the minimum pseudo-inverse $f_j^{(-1)}(A)$ is computable in a constant time for any A and $j \in [1, n]$,
- an $O(n \log_2 \delta)$ time algorithm for the problem K1B, provided that the profit function $f_j(x_j)$ is non-negative non-decreasing and the difference between any two distinct values $f_j(x_j)$, $j \in [1, n]$, is at least $\frac{1}{\delta}$ for a positive integer δ, and that the value $f_j^{(-1)}(A)$ is computable in a constant time for any A and $j \in [1, n]$,

- an $O(n)$ time algorithm for the problem K2B, provided that the profit function $f_j(x_j)$ is non-negative non-decreasing and the value $f_j^{(-1)}(E)$ is computable in a constant time for any $j \in [1, n]$,
- proof of NP-hardness of the problems K1 and K2 with arbitrary non-negative non-decreasing profit functions, and
- FPTASes with running times $O(n^3 \log_2 n + \frac{n^3}{\varepsilon^2})$ and $O(n^3 \log_2(nd_{\max}) + \frac{n^3}{\varepsilon^2})$ for the problems K1 and K2, respectively, provided that the profit function $f_j(x_j)$ is arbitrary non-negative non-decreasing and the value $f_j^{(-1)}(A)$ is computable in a constant time for any A and $j \in [1, n]$.

For future research, it is worth to establish computational complexity of the problems K1 and K2 with concave non-decreasing profit functions and to study theoretically and practically interesting modifications of K1 and K2.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Appendix. Bound improvement procedure

The procedure calculates numbers B and C such that $B \leq F^* \leq C$ and $C \leq 4B$. For a maximization problem with optimum value F^*, let lower and upper bounds V' and U' be given such that $0 < V' \leq F^* \leq U'$. Compute integer number k satisfying $2^{k-1}V' < U' \leq 2^k V'$. We have $V' \leq F^* \leq 2^k V'$ and $k = O\left(\log_2 \frac{U'}{V'}\right)$. For $q = k, k-1, \ldots, 1$, apply an approximation algorithm for the maximization problem which finds a feasible solution with value $F^{(q)} \geq F^* - 2^{q-2}V'$ if $F^* \leq 2^q V'$. If $F^* > 2^q V'$, then the algorithm can still find a feasible solution with value $F^{(q)} \geq F^* - 2^{q-2}V'$, but it is not guaranteed. For the problem KG1, algorithm DP-R1 with $\varepsilon = 1/2$, $U = 2^q V'$ and $V = 2^{q-1}V'$ is such an algorithm because it finds a solution with value $F^0 \geq F^* - \varepsilon V = F^* - \frac{1}{2}V = F^* - 2^{q-2}V'$ if $F^* \leq U = 2^q V'$. For the problem KG1, it runs in $O(n^3)$ time. Since $F^* \leq 2^k V'$, a feasible solution with value $F^{(k)} \geq F^* - 2^{k-2}V'$ will be found. If $F^{(k)} \geq 2^{k-2}V'$, then $2^{k-2}V' \leq F^{(k)} \leq F^* \leq 2^k V'$. Hence, $B = F^{(k)}$, $C = 2^k V'$, and the procedure stops. If $F^{(k)} < 2^{k-2}V'$, then $F^* \leq F^{(k)} + 2^{k-2}V' < 2^{k-1}V'$, and we can re-set $k := k-1$. Therefore, $B = F^{(t)}$ and $C = 2^t V'$, where t is the largest index such that $k \geq t \geq 1$ and $F^{(t)} \geq 2^{t-2}V'$. Such an index t exists. Assume that it does not exist for all iterations up to the iteration in which k is re-set to 1. Since this iteration is reached, relations $F^{(2)} < 2^0 V'$ and $F^* \leq F^{(2)} + 2^0 V' < 2L'$ are satisfied. Furthermore, $V' \leq F^*$ by the definition of V'. Hence, $V' \leq F^* \leq 2V'$, and for $t = 1$ we have $F^{(1)} \geq F^* - 2^{-1}V' \geq 2^{-1}L'$, as it is required by the definition of t. Note that if $t = 1$, then $C/B = 2$. For the problem KG1, since $0 < V \leq F^* \leq nV$, the number of iterations of the bound improvement procedure is at most $O(\log_2 n)$ and its overall running time is $O(n^3 \log_2 n)$.

References

Balas E, Zemel E (1980) An algorithm for large zero-one knapsack problems. Oper Res 28(5):1130–1154

Blum M, Floyd RW, Pratt V, Rivest RL, Tarjan RE (1973) Time bounds for selection. J Comput Syst Sci 7(4):448–461

Bretthauer KM, Shetty B (2002) The nonlinear knapsack problem – algorithms and applications. European J Oper Res 138(3):459–472

D'Ambrosio C, Martello S (2011) Heuristic algorithms for the general nonlinear separable knapsack problem. Comput Oper Res 38(2):505–513

D'Ambrosio C, Martello S, Mencarelli L (2018) Relaxations and heuristics for the multiple non-linear separable knapsack problem. Comput Oper Res 93:79–89

Dantzig GB (1957) Discrete-variable extremum problems. Oper Res 5(2):266–277

Garey MR, Johnson DS (1979) Comput Intractability: A Guide to the Theory of NP-Completeness. W. H, Freeman and Company, San Francisco

Gurevsky E, Kopelevich D, Kovalev S, Kovalyov MY (2022) Min-sum controllable risk problems with concave risk functions of the same value range. Networks 79(1):105–116

Halman N, Holzhauser M, Krumke SO (2018) An FPTAS for the knapsack problem with parametric weights. Oper Res Lett 46(5):487–491

Ibarra OH, Kim CE (1975) Fast approximation algorithms for the knapsack and sum of subset problems. J ACM 22(4):463–468

Kellerer H, Pferschy U, Pisinger D (2004) Knapsack Probl. Springer-Verlag, Berlin Heidelberg

Kovalev S (2021) Approximation issues of fractional knapsack with penalties: a note. 4OR: A Quart J Oper Res. https://doi.org/10.1007/s10288-021-00474-1

Kovalyov MY (1995) Improving the complexities of approximation algorithms for optimization problems. Oper Res Lett 17(2):85–87

Malaguti E, Monaci M, Paronuzzi P, Pferschy U (2019) Integer optimization with penalized fractional values: The Knapsack case. Eur J Oper Res 273(3):874–888

Martello S, Toth P (1990) Knapsack Probl: Algorithms Comput Implementations. John Wiley & Sons, Chichester

Sahni S (1977) General techniques for combinatorial approximation. Oper Res 25(6):920–936

Zhang B, Chen B (2012) Heuristic and exact solution method for convex nonlinear knapsack problem. Asia-Pacific J Oper Res 29(05):1250031