4OR (2022) 20:209–216
https://doi.org/10.1007/s10288-021-00474-1

RESEARCH PAPER

Approximation issues of fractional knapsack with penalties: a note

Sergey Kovalev[1]

Received: 27 November 2020 / Revised: 27 November 2020 / Accepted: 21 January 2021 /
Published online: 23 February 2021

Abstract

Malaguti et al. introduce (Eur J Oper Res 273:874–888, 2019) the Fractional Knapsack Problem with Penalties, which is similar to the classical 0-1 Knapsack problem, except that each of the n variables associated with one of the n items can take any value from the interval [0, 1], and values other than 0 and 1 are penalized. They state that the problem is NP-hard in the ordinary sense as a generalization of the classical 0-1 knapsack problem and develop a fully polynomial-time approximation scheme for the case of non-negative non-decreasing profit functions. It is demonstrated that, unless $\mathcal{P} = \mathcal{NP}$, no polynomial-time approximation algorithm with any approximation ratio exists for the case of polynomially defined, polynomially computable, discontinuous and non-monotone penalty functions even if there is a single item. A fully polynomial-time approximation scheme which is roughly n times faster than the one of Malaguti et al. for the same case is also presented.

Keywords Knapsack problem · Computational complexity · Fully polynomial-time approximation scheme

Mathematics Subject Classification 90–10

1 Introduction

Malaguti et al. (2019) study the following Fractional Knapsack Problem with Penalties (FKPP). There are items of a set $N = \{1, \ldots, n\}$ and a knapsack with weight capacity $W > 0$. Any part of an item can be packed into the knapsack. Let variable x_j, $0 \leq x_j \leq 1$, represent the part of the item j packed into the knapsack. Each item $j \in N$ is associated with a weight $w_j > 0$, a *full-size-item* profit $p_j > 0$ and a profit function

✉ Sergey Kovalev
skovalev@inseec.com

[1] INSEEC School of Business and Economics, INSEEC U. Research Center, 25 rue de l'Université, 69007 Lyon, France

$$F_j(x_j) = \begin{cases} 0 & \text{if } x_j = 0, \\ p_j & \text{if } x_j = 1, \\ p_j x_j - f_j(x_j) & \text{if } 0 < x_j < 1, \end{cases}$$

where $f_j(x_j)$ is an arbitrary, even discontinuous, non-negative penalty function applied when a fractional part of the item j is packed into the knapsack. Denote $x = (x_1, \dots, x_n)$. The FKPP problem is

$$\max_x \sum_{j \in N} F_j(x_j), \text{ subject to}$$
$$\sum_{j \in N} w_j x_j \le W,$$
$$0 \le x_j \le 1, \, j \in N.$$

Note that, in the classical knapsack problem, relations $w_j \le W$, $j \in N$, can be assumed, because if $w_j > W$, then item j cannot be packed into the knapsack, and hence, it can be removed from the input. In FKPP, if $w_j > W$, then a part of item j can still be packed into the knapsack, and therefore, relations $w_j \le W$, $j \in N$, are restrictive. Assume that in FKPP weights w_j, $j \in N$, are not bounded from above by W.

Let ε be a number or a function of the input data of FKPP such that $0 < \varepsilon < 1$. An algorithm for a maximization problem is called an $(1-\varepsilon)$-*approximation algorithm* if for any instance of the problem it finds a feasible solution with value $H^0 \ge H^* - \varepsilon|H^*|$, where H^* is the maximum value. The absolute value of H^* is used in this definition because H^* can be negative. A *Fully Polynomial-Time Approximation Scheme (FPTAS)* for a maximization problem is a collection of $(1 - \varepsilon)$-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.

Malaguti et al. (2019) remark that FKPP is an NP-hard problem since the classical 0-1 knapsack problem reduces to it by setting $f_j(x_j) = p_j x_j$, $j \in N$. They concentrate on a special case of FKPP, in which $F_j(x_j)$, $j \in N$, are non-negative non-decreasing functions. For this case, among other results, they develop an FPTAS with running time $O(n^4/\varepsilon^2)$.

Earlier publications, in which optimization problems with penalized item splitting are studied, include Freling et al. (2003), Liu and Cheng (2004), Lodi et al. (2011), Archetti and Speranza (2012), Malaguti et al. (2014), Casazza and Ceselli (2014), Fung (2014) and Liu and Draper (2016).

In the next section, it is proved that, unless $\mathcal{P} = \mathcal{NP}$, no polynomial-time $(1 - \varepsilon)$-approximation algorithm exists for FKPP even if $n = 1$ and the penalty function $f_1(x_1)$ is polynomially defined, polynomially computable, discontinuous and non-monotone. It is shown that this statement also applies for the general FKPP with an arbitrary number of items. Section 3 contains an $O(n^3 \log_2 n + n^3/\varepsilon^2)$ time FPTAS for FKPP with non-negative non-decreasing profit functions.

2 Non-approximability

The non-approximability proof is similar to the NP-hardness proof of Cheng and Kovalyov (2002) for an unconstrained optimization problem.

Theorem 1 *Unless $\mathcal{P} = \mathcal{NP}$, no polynomial-time $(1-\varepsilon)$-approximation algorithm with any $0 < \varepsilon < 1$ exists for FKPP even if $n = 1$ and the penalty function $f_1(x_1)$ is polynomially computable, polynomially defined, discontinuous and non-monotone.*

Proof Firstly, it is shown that the decision version of FKPP in the statement of the theorem is NP-complete by a transformation from the NP-complete problem Partition (Garey and Johnson 1979): Given positive integers $a_1, \dots, a_k$ and A such that $\sum_{i=1}^k a_i = 2A$, is there a 0-1 vector $y = (y_1, \dots, y_k\}$, $y \notin \{(0, \dots, 0), (1, \dots, 1)\}$, such that $\sum_{i=1}^k a_i y_i = A$?

Given any instance of Partition, construct the following instance of FKPP. Set $n = 1$, $p_1 = 2(2^k - 1)$, $w_1 = 2$ and $W = 1$. Let $b(z)$ denote a binary representation of z, $z \in \{1, 2, \dots, \frac{p_1}{2}\}$, using k bits, $b(z) = (b_1(z), \dots, b_k(z))$, $b_i(z) \in \{0, 1\}$, $i = 1, \dots, k$, such that $z = \sum_{i=1}^k 2^{k-i} b_i(z)$. Define

$$f_1(x_1) = \begin{cases} p_1 x_1 - 1 + |\sum_{i=1}^k a_i b_i(\lfloor p_1 x_1 \rfloor) - A| & \text{if } x_1 \in \{\frac{q}{p_1} \mid q = 1, \dots, \frac{p_1}{2}\} \\ p_1 x_1 & \text{if } x_1 \in \{\frac{q}{p_1} \mid q = \frac{p_1}{2} + 1, \dots, p_1\}. \end{cases}$$

Domain of the function $f_1(x_1)$ is the discrete set $\{\frac{q}{p_1} \mid q = 1, \dots, p_1\}$. It is easy to see that the function $p_1(x_1)$ is polynomially defined, polynomially computable, discontinuous and non-monotone. Furthermore, the length of p_1 in binary encoding is $O(k)$. The value $\lfloor p_1 x_1 \rfloor$ and the binary representation $b(\lfloor p_1 x_1 \rfloor)$ for any $x_1 \in \{\frac{q}{p_1} \mid q = 1, \dots, \frac{p_1}{2}\}$ can be found in $O(k)$ time. Using $b(\lfloor p_1 x_1 \rfloor)$, the value $f_1(x_1)$ can be calculated in $O(k)$ time as well. Therefore, the reduction is polynomial with respect to the input length of the Partition instance. Clearly, $f_1(x_1) \geq 0$ for $0 < x_1 < 1$, as it is required by the definition of FKPP.

Since $f_1(x_1)$ is polynomially computable and defined for $x_1 \in \{\frac{q}{p_1} \mid q = 1, \dots, p_1\}$, the variable value of q in $x_1 = \frac{q}{p_1}$ can be taken as a certificate to verify whether a decision version of FKPP has a solution. The length of such a certificate is bounded by a polynomial of $\log p_1$. It is deduced that a decision version of FKPP is in the class $\mathcal{NP}$. The transformation is the following.

It is shown that an instance of Partition has a solution if and only if for the associated instance of FKPP there exists a solution $x_1 \in \{\frac{q}{p_1} \mid q = 1, \dots, p_1\}$ such that $F_1(x_1) \geq 1$ and $w_1 x_1 \leq W$. Observe that the definitions of the functions $f_1(x_1)$ and $F_1(x_1)$ imply that the values $F_1(x_1)$ are integral. Assume that the instance of Partition has a solution $y = (y_1, \dots, y_k)$, $y_i \in \{0, 1\}$, $i = 1, \dots, k$. Define $z = \sum_{i=1}^k 2^{k-i} y_i \geq 1$ and $x_1 = \frac{z}{p_1}$. Observe that $1 \leq p_1 x_1 = z \leq 2^k - 1 = \frac{p_1}{2}$.

Hence, $x_1 \in \{\frac{q}{p_1} \mid q = 1, \dots, \frac{p_1}{2}\}$ and $w_1 x_1 = 2x_1 \le 1 = W$. Furthermore,

$$F_1(x_1) = 1 - \Big|\sum_{i=1}^{k} a_i b_i(\lfloor p_1 x_1 \rfloor) - A\Big| = 1 - \Big|\sum_{i=1}^{k} a_i b_i(z) - A\Big|$$
$$= 1 - \Big|\sum_{i=1}^{k} a_i y_i - A\Big| = 1.$$

It was shown that the constructed instance of FKPP has a solution if the associated instance of Partition has a solution. Now, assume that the instance of FKPP has a solution $x_1 \in \{\frac{q}{p_1} \mid q = 1, \dots, p_1\}$ such that $F_1(x_1) \ge 1$ and $w_1 x_1 \le W$. These relations imply that $x_1 \in \{\frac{q}{p_1} \mid q = 1, \dots, \frac{p_1}{2}\}$, $\lfloor p_1 x_1 \rfloor \le p_1/2 \le 2^k - 1$ and $F_1(x_1) = 1 - |\sum_{i=1}^{k} a_i b_i(\lfloor p_1 x_1 \rfloor) - A| \ge 1$. The latter relation can only happen if $\sum_{i=1}^{k} a_i b_i(\lfloor p_1 x_1 \rfloor) = A$. It is deduced that the 0-1 vector y such that $y_i = b_i(\lfloor p_1 x_1 \rfloor)$, $i = 1, \dots, k$, is a solution of the associated instance of Partition.

Thus, FKPP is NP-hard if $n = 1$ and the penalty function $f_1(x_1)$ is polynomially computable, polynomially defined, discontinuous and non-monotone. Assume that $0 < \varepsilon < 1$ and there exists a $(1-\varepsilon)$-approximation algorithm for FKPP. Let it deliver a solution x_1^0 with value F_1^0 for the FKPP instance associated with a Partition instance. Since $w_1 x_1^0 = 2x_1^0 \le W = 1$, we know that $x_1 \in \{\frac{q}{p_1} \mid q = 1, \dots, \frac{p_1}{2}\}$. Together with the definitions of the functions $f_1(x_1)$ and $F_1(x)$, this implies that F_1^0 is an integer satisfying $F_1^0 \le 1$. Denote by F_1^* the maximum objective function value for the instance of FKPP. Similar to F_1^0, the number F_1^* is an integer satisfying $F_1^* \le 1$. By the definition of an $(1-\varepsilon)$-approximation algorithm, relations

$$F_1^* - \varepsilon|F_1^*| \le F_1^0 \le F_1^* \tag{1}$$

must be satisfied.

It is shown in the above NP-completeness proof that if $F_1^* = 1$, then the Partition instance has a solution, and if $F_1^* \le 0$, then the Partition instance has no solution. It follows from (1), relation $0 < \varepsilon < 1$ and integrality of F_1^0 and F_1^* that $F_1^0 = 1$ if and only if $F_1^* = 1$. It is deduced that $F_1^0 = 1$ if and only if the Partition instance has a solution. Thus, if a polynomial-time $(1-\varepsilon)$-approximation algorithm for FKPP exists, then Partition admits a polynomial-time algorithm, which is impossible, unless $\mathcal{P} = \mathcal{NP}$. □

Corollary 1 *Unless $\mathcal{P} = \mathcal{NP}$, no polynomial-time $(1-\varepsilon)$-approximation algorithm with any $0 < \varepsilon < 1$ exists for FKPP.*

Proof For any instance I_1 of FKPP in the proof of Theorem 1, construct an instance I_2 of the general FKPP, in which $W = 1$, $n = \log_2 p_1$, all the input parameters of item 1 are the same as in I_1, $w_j = 0$ and $f_j(x_j) = p_j = 0$ for $0 \le x_j \le 1$, $j = 2, \dots, n$. The transformation of I_1 into I_2 is polynomial in the input length of I_1. The remaining proof is similar to the proof of Theorem 1. □

3 Non-negative non-decreasing profit functions: FPTAS

Consider FKPP with non-negative non-decreasing profit functions $F_j(x_j)$, $j \in N$. Similar to the FPTAS of Malaguti et al. (2019), the present FPTAS is based on a scaled and rounded re-formulation of FKPP and a dynamic programming algorithm for the latter problem. However, the current re-formulation is different from that in Malaguti et al. (2019), which results in a more efficient running time.

Let $x^* = (x_1^*, \dots, x_n^*)$ be an optimal solution of FKPP and $F^* = \sum_{j \in N} F_j(x_j^*)$. Observe that any 0-1 vector $x^{(j)}$ such that $x_j^{(j)} = \min\{1, \frac{W}{w_j}\}$ and $x_i^{(j)} = 0$, $i \neq j$, $i \in N$, $j \in N$, is feasible for FKPP. Hence, $L' = \max_{j \in N}\{F_j(\min\{1, \frac{W}{w_j}\})\}$ is a lower bound for F^*. Furthermore, if $F_j(\min\{1, \frac{W}{w_j}\}) = 0$, then, since F_j is non-negative and non-decreasing, the contribution of item j to the objective function is zero in any feasible solution, and this item can be removed from the input. Therefore, assume without loss of generality that $F_j(\min\{1, \frac{W}{w_j}\}) > 0$, $j \in N$. One more observation is that $F_j(\min\{1, \frac{W}{w_j}\})$ is the maximum contribution of item j to the objective function value of any feasible solution because the functions $F_j(x_j)$ are non-decreasing, $j \in N$. It follows that $U' = nL'$ is an upper bound on F^*.

From now on assume that arbitrary lower and upper bounds L and U are known such that $0 < L' \leq L \leq F^* \leq U \leq U'$. For a given $\varepsilon > 0$, define scaling parameter $\delta = \frac{\varepsilon L}{n}$ and formulate the following *Scaled and Rounded FKPP (SRK)* problem.

$$\max_x \sum_{j \in N} \left\lfloor \frac{F_j(x_j)}{\delta} \right\rfloor, \text{ subject to}$$

$$\sum_{j \in N} w_j x_j \leq W,$$

$$x_j \in \{r_j(0), r_j(1), \dots, r_j(\left\lfloor \frac{U}{\delta} \right\rfloor)\}, \; j \in N,$$

where

$$r_j(s) = \inf \left\{ x \mid 0 \leq x \leq \min\left\{1, \frac{W}{w_j}\right\}, \left\lfloor \frac{F_j(x)}{\delta} \right\rfloor = s \right\} =$$

$$\inf \left\{ x \mid 0 \leq x \leq \min\left\{1, \frac{W}{w_j}\right\}, s\delta \leq F_j(x) < (s+1)\delta \right\}, \; j \in N.$$

Note that, by the definition of the infimum, the value $r_j(s)$ may not exist. Assume that the value $\inf\{x \mid F_j(x) \geq D\}$ can be computed in $O(1)$ time for any D, $0 < D \leq U$, and $j \in N$. Then, since functions $F_j(x)$ are non-decreasing, all the existing values $r_j(s)$ for $s = 0, 1, \dots, \lfloor \frac{U}{\delta} \rfloor$, $j \in N$, can be computed in $O(\frac{nU}{\delta})$ time. A similar assumption is made by Malaguti et al. (2019).

Let x^0, S^0 and F^0 denote an optimal solution of SRK, its objective function value with respect to SRK and its objective function value with respect to the original FKPP, respectively. Note that x^0 is feasible for FKPP. Furthermore, there exists a feasible solution x' of SRK such that $\lfloor \frac{F_j(x'_j)}{\delta} \rfloor = \lfloor \frac{F_j(x^*_j)}{\delta} \rfloor$, $j \in N$. Let us evaluate the quality of x^0 with respect to FKPP.

$$F^0 = \sum_{j \in N} F_j(x^0_j) = \delta \sum_{j \in N} \frac{F_j(x^0_j)}{\delta} \geq \delta \sum_{j \in N} \left(\left\lfloor \frac{F_j(x^0_j)}{\delta} \right\rfloor \right) \geq \delta \sum_{j \in N} \left(\left\lfloor \frac{F_j(x'_j)}{\delta} \right\rfloor \right) =$$
$$\delta \sum_{j \in N} \left(\left\lfloor \frac{F_j(x^*_j)}{\delta} \right\rfloor \right) > \delta \sum_{j \in N} \left(\frac{F_j(x^*_j)}{\delta} - 1 \right) = F^* - n\delta \geq (1 - \varepsilon) F^*.$$

This chain of relations proves that any optimal algorithm for SRK is an $(1 - \varepsilon)$-approximation algorithm for FKPP. Let us evaluate S^0 from above.

$$S^0 = \sum_{j \in N} \left\lfloor \frac{F_j(x^0_j)}{\delta} \right\rfloor \leq \sum_{j \in N} \frac{F_j(x^0_j)}{\delta} \leq \sum_{j \in N} \frac{F_j(x^*_j)}{\delta} \leq \frac{U}{\delta}.$$

Let $U^0 = \lfloor \frac{U}{\delta} \rfloor$. Since S^0 is integral, $S^0 \in \{0, 1, \ldots, U^0\}$.

A profit-based dynamic programming algorithm for SRK, denoted as DP, is now presented. In this algorithm, partial solutions $x = (x_1, \ldots, x_j)$ are constructed for $j = 1, \ldots, n$. With each partial solution $x = (x_1, \ldots, x_j)$, a state (j, g) is associated, where $g = \sum_{i=1}^{j} \lfloor \frac{F_i(x_i)}{\delta} \rfloor$. A total weight function $T_j(g)$ is recursively calculated for each state (j, g), which is the minimum total weight of partial solutions in the same state (j, g). A partial solution with value $T_j(g)$ is dominant over partial solutions in the state (j, g) meaning that if there is a partial solution in this state that can be extended to an optimal solution of SRK, then the dominant solution can be extended to an optimal solution of SRK in the same way. Therefore, only a single dominant solution needs to be stored for each state, and the other solutions can be discarded.

The initialization of Algorithm DP is $T_0(0) = 0$, and $T_0(g) = W + 1$ for $g \neq 0$, $g = 0, 1, \ldots, U^0$. The recursion for $j = 1, \ldots, n$ and $g = 0, 1, \ldots, U^0$, is

$$T_j(g) = \min_{x \in \{r_j(0), r_j(1), \ldots, r_j(U^0)\}} \left\{ T_{j-1}\left(g - \left\lfloor \frac{F_j(x)}{\delta} \right\rfloor \right) + w_j x \right\}.$$

The optimal objective function value of the problem SRK is equal to

$$S^0 = \max\{g \mid T_n(g) \leq W, g = 0, 1, \ldots, U^0\},$$

and the corresponding optimal solution x^0 is determined by the optimal solutions of the recursive equation on the way to $T_n(g) = S^0$. The recursive equation can be solved in $O(\frac{U}{\delta})$ time for each state (j, g). Since the state space cardinality is $O(nU^0) = O(\frac{nU}{\delta})$, the running time of Algorithm DP is $O(\frac{nU^2}{\delta^2}) = O(\frac{n^3}{\varepsilon^2} (\frac{U}{L})^2)$.

Remark 1 Algorithm DP produces a solution with profit $F^0 \geq F^* - \varepsilon L$ and it does it in $O(\frac{n^3}{\varepsilon^2}(\frac{U}{L})^2)$ time irrespectively of whether L, $0 < L' \leq L \leq U$, is a lower bound for F^*.

Kovalyov (1995) presents a bound improvement procedure for a minimization problem, which can be easily modified for a maximization problem, to determine lower and upper bounds B and C for optimum value F^* such that $B \leq F^* \leq C$ and $B/C = O(1)$, if there exist lower and uppers bounds L' and U' satisfying $0 < L' \leq F^* \leq U'$ and a specific approximation algorithm. For FKPP, DP is such an algorithm. The running time of the bound improvement procedure for the FKPP problem is $O(n^3 \log_2 n)$ if it employs lower and upper bounds L' and $U' = nL'$ and Algorithm DP. For completeness, the bound improvement procedure after the following lemma is presented.

Lemma 1 *A combination of Algorithm DP and the bound improvement procedure is an FPTAS for FKPP with running time $O(n^3 \log_2 n + \frac{n^3}{\varepsilon^2})$, if the profit functions $F_j(x)$, $j \in N$, are non-negative non-decreasing and the value* $\inf\{x \mid F_j(x) \geq D\}$ *is computable in $O(1)$ time for any D, $0 < D \leq U$, and $j \in N$.*

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 the optimum value F^*, let lower and upper bounds L' and U' be given such that $0 < L' \leq F^* \leq U'$. Compute integer number k satisfying $2^{k-1}L' < U' \leq 2^k L'$. We have $L' \leq F^* \leq 2^k L'$ and $k = O(\log_2 \frac{U'}{L'})$.

For $q = k, k-1, \dots, 1$, apply an approximation algorithm for the maximization problem which finds a feasible solution with value $F^{(q)} \geq F^* - 2^{q-2}L'$ if $F^* \leq 2^q L'$. If $F^* > 2^q L'$, then the algorithm can still find a feasible solution with value $F^{(q)} \geq F^* - 2^{q-2}L'$, but it is not guaranteed. For FKPP, Algorithm DP with $\varepsilon = 1/2$, $U = 2^q L'$ and $L = 2^{q-1}L'$ is such an algorithm because, due to Remark 1, it finds a solution with value $F^0 \geq F^* - \varepsilon L = F^* - \frac{1}{2}L = F^* - 2^{q-2}L'$ if $F^* \leq U = 2^q L'$. For FKPP, it runs in $O(n^3)$ time. Since $F^* \leq 2^k L'$, a feasible solution with value $F^{(k)} \geq F^* - 2^{k-2}L'$ will be found. If $F^{(k)} \geq 2^{k-2}L'$, then $2^{k-2}L' \leq F^{(k)} \leq F^* \leq 2^k L'$. Hence, $B = F^{(k)}$, $C = 2^k L'$, and the procedure stops. If $F^{(k)} < 2^{k-2}L'$, then $F^* \leq F^{(k)} + 2^{k-2}L' < 2^{k-1}L'$, and we can re-set $k := k-1$. Therefore, $B = F^{(t)}$ and $C = 2^t L'$ where t is the largest index such that $k \geq t \geq 1$ and $F^{(t)} \geq 2^{t-2}L'$. 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 $F^{(2)} < 2^0 L'$ and $F^* \leq F^{(2)} + 2^0 L' < 2L'$ are satisfied. Furthermore, $L' \leq F^*$ by the definition of L'. Hence, $L' \leq F^* \leq 2L'$, and for $t = 1$ we have $F^{(1)} \geq F^* - 2^{-1}L' \geq 2^{-1}L'$, as it is required by the definition of t. Note that if $t = 1$, then $C/B = 2$. For FKPP, since $0 < L' \leq F^* \leq nL'$, the number of iterations of the bound improvement procedure is at most $k = O(\log_2 \frac{U'}{L'}) = O(\log_2 n)$ and its overall running time is $O(n^3 \log n)$.

Compliance with ethical standards

Conflict of interest The author declares that he has no conflict of interest.

References

Archetti C, Speranza MG (2012) Vehicle routing problems with split deliveries. Int Trans Oper Res 19(1–2):3–22

Casazza M, Ceselli A (2014) Mathematical programming algorithms for bin packing problems with item fragmentation. Comput Oper Res 46:1–11

Cheng TCE, Kovalyov MY (2002) An unconstrained optimization problem is NP-hard given an oracle representation of its objective function: a technical note. Comput Oper Res 29:2087–2091

Freling R, Romeijn E, Morales DR, Wagelmans A (2003) A branch and price algorithm for the multi-period single-sourcing problem. Oper Res 51(6):922–939

Fung SPY (2014) Online scheduling with preemption or non-completion penalties. J Sched 17(2):173–183

Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. W.H. Freeman and Co., San Francisco

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

Liu Z, Cheng TCE (2004) Minimizing total completion time subject to job release dates and preemption penalties. J Sched 7(4):313–327

Liu X, Draper SC (2016) The ADMM penalized decoder for LDPC codes. IEEE Trans Inf Theory 62(6):2966–2984

Lodi A, Martello S, Monaci M, Cicconetti C, Lenzini L, Mingozzi E, Eklund C, Moilanen J (2011) Efficient two-dimensional packing algorithms for mobile WiMAX. Manag Sci 57:2130–2144

Malaguti E, Durán RM, Toth P (2014) Approaches to real world two-dimensional cutting problems. Omega 47:99–115

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