Reductions between scheduling problems with non-renewable resources and knapsack problems

Péter Györgyi[a], Tamás Kis[b,*]

[a]*Department of Operations Research, Loránd Eötvös University, H1117 Budapest, Pázmány Péter sétány 1/C, Hungary*
[b]*Institute for Computer Science and Control, Hungarian Academy of Sciences, H1111 Budapest, Kende str. 13–17, Hungary*
Tel.: +36 1 2796156; Fax: +36 1 4667503

Abstract

In this paper we establish approximation preserving reductions between scheduling problems in which jobs either consume some raw materials, or produce some intermediate products, and variants of the knapsack problem. Through the reductions, we get new approximation algorithms, as well as inapproximability results for the scheduling problems.

Keywords: Approximation preserving reductions, scheduling problems, knapsack problems

1. Introduction

In this paper we study approximation preserving reductions between single machine scheduling problems extended with non-renewable resources, and various knapsack problems. We will consider two types of scheduling problems: (i) scheduling of jobs *producing* some intermediate products, and (ii) scheduling of jobs *consuming* some raw materials. In the former case, the jobs produce intermediate products to meet demands at given dates, whereas in the second case, jobs consume raw materials whose stock is replenished at given dates and in known quantities. On the other hand, we will consider two variants of the knapsack problem. Beside the *basic knapsack problem*, in which there is a set of items each having a size and a profit, and a subset of items of maximum profit, but of limited total size must be chosen, we will also consider the *multi-dimensional knapsack problem* in which the knapsack has sizes in multiple dimensions.

Approximation preserving reductions are useful for obtaining both positive and negative results. Consider, say, the PTAS reduction, which reduces an optimization problem Π_1 to another optimization problem Π_2 in such a manner

*Corresponding author
Email addresses: gyorgyipeter@gmail.com (Péter Györgyi), tamas.kis@sztaki.mta.hu (Tamás Kis)

that if there is a PTAS for Π_2, then this yields a PTAS for Π_1 as well (for formal definitions, see Section 3). So, we can get a positive result for an optimization problem Π_1, i.e., a PTAS, if we can identify another optimization problem Π_2 which admits a PTAS, and if we manage to devise a PTAS reduction from Π_1 to Π_2. On the other hand, if we want to prove that some problem Π_2 does not admit a PTAS unless $\mathcal{P} = \mathcal{NP}$, it suffices to find another optimization problem Π_1 which does not admit a PTAS unless $\mathcal{P} = \mathcal{NP}$, and a PTAS reduction from Π_1 to Π_2. Among the many types of reductions published in the literature, we will only use the PTAS-, the FPTAS- and the Strict-reductions (see Section 3).

Before we proceed we provide a more formal definition of those problems studied in this paper.

1.1. Knapsack Problems

In the (basic) *Knapsack Problem* (KP) there is a set of n items j with profit v_j and weight w_j. One has to select a subset of the items with the largest total profit so that the total weight of the selected items is at most a given constant ('capacity') b'. Formally:

$$OPT_{KP} := \max \sum_{j=1}^{n} v_j x_j \tag{1}$$

$$\sum_{j=1}^{n} w_j x_j \leq b' \tag{2}$$

$$x_j \in \{0,1\}, \quad j = 1, \ldots, n. \tag{3}$$

We will use the notation OPT_{KP} for the optimal value of this problem.

In the *r-dimensional Knapsack Problem (r-DKP)* each item has r weights and there are r constraints:

$$OPT_{r-DKP} := \max \sum_{j=1}^{n} v_j x_j \tag{4}$$

$$\sum_{j=1}^{n} w_{ij} x_j \leq b'_i, \quad i = 1, \ldots, r \tag{5}$$

$$x_j \in \{0,1\}, \quad j = 1, \ldots, n. \tag{6}$$

The optimum value of this problem is denoted by OPT_{r-DKP}.

1.2. Resource Scheduling Problems

In this section we recapitulate two resource scheduling problems, the *Delivery tardiness problem* (see [10]) and the *Material consumption problem* (see e.g. [5],[15]).

In the *Delivery tardiness problem (DTP_q^r)* there are a single machine, a finite set of n jobs, and a set of r materials produced by the jobs. The machine can perform only one job at a time, and preemption is not allowed. Job J_j, $j \in$

$\{1,\ldots,n\}$, has a processing time $p_j \in \mathbb{Z}_+$, and produces some materials, which is described by an r-dimensional non-negative vector $a_j \in \mathbb{Z}^r_+$. There are due dates along with required shipments, i.e., pairs (u_ℓ, b_ℓ) with $u_\ell \in \mathbb{Z}_+$, and $b_\ell \in \mathbb{Z}^r_+$, $\ell = 1,\ldots,q$, and $0 \le u_1 < \cdots < u_q$. The solution of the problem is a sequence σ of the jobs. The starting time of the i^{th} job is then $S_{\sigma(i)} = \sum_{k=1}^{i-1} p_{\sigma(k)}$. A shipment (u_ℓ, b_ℓ) is *met* by S, if the total production of those jobs finishing by u_ℓ is at least $\tilde{b}_\ell := \sum_{k=1}^{\ell} b_k$, i.e., $\sum_{(j\,:\,S_j+p_j\le u_\ell)} a_j \ge \tilde{b}_\ell$ (coordinate wise), otherwise it is *tardy*. Let $C_\ell(S)$ be the earliest time point $t \ge 0$ with $\sum_{(j\,:\,S_j+p_j\le t)} a_j \ge \tilde{b}_\ell$. The *tardiness* of a shipment is $T_\ell(S) := \max\{0, C_\ell(S) - u_\ell\}$. The *maximum tardiness of a schedule* is $T_{\max}(S) := \max_\ell T_\ell(S)$. The objective is to minimize the maximum tardiness. We denote this problem by $1|dm = r|T_{\max}$, where '$dm = r$' indicates that the number of products is fixed to r (not part of the input). An important special case of this problem is when there are only two time points ($0 \le u_1 < u_2$) when some product is due (denoted by $1|dm = r, q = 2|T_{\max}$). Since $T_{\max}$ can be 0 in an optimal solution, we will consider the *shifted delivery tardiness objective function* defined as $T^s_{\max} := T_{\max} + \mathit{const}$, where *const* is a positive constant, depending on the problem data.

In the *Material consumption problem* (MCP^r_q) there are a single machine, a finite set of n jobs, and a set of r materials consumed by the jobs. The machine can perform only one job at a time, and preemption is not allowed. There are n jobs J_j, $j = 1,\ldots,n$, each characterized by two numbers: processing time p_j and quantities consumed from the resources $a_j \in \mathbb{Z}^r_+$. The resources have initial stocks, and they are replenished at given moments in time, i.e., there are q pairs $(u_1, b_1),\ldots,(u_q, b_q)$, with $0 = u_1 < \cdots < u_q$ being the time points and the $b_\ell \in \mathbb{Z}^r_+$ the quantities supplied. A *schedule* S specifies a starting time for each job such that the jobs do not overlap in time, and the total material supply up to the starting time of every job is at least the total request of those jobs starting not later than S_j, i.e., $\sum_{(\ell\,:\,u_\ell\le S_j)} b_\ell \ge \sum_{(j'\,:\,S_{j'}\le S_j)} a_{j'}$ (coordinate wise). The objective is to minimize the *makespan* defined as the maximum job completion time. We denote this problem by $1|rm = r|C_{\max}$, where '$rm = r$' indicates that the number of the raw materials is fixed to r (not part of the input). An important special case of this problem is when there are only two time points ($u_1 = 0$ and $u_2 > 0$) when some resource is supplied ($1|rm = r, q = 2|C_{\max}$).

Assumption 1. *In both problems $\sum_\ell b_\ell = \sum_j a_j$ holds without loss of generality.*

The notation used throughout the paper is summarized in Appendix A.

1.3. Results

Our goal in this paper is to systematically examine reducibility relations between knapsack problems and scheduling problems with consumer or producer jobs. Our main results are approximation preserving reductions among three problem classes: (i) special cases of scheduling problems with producer jobs, (ii) special cases of scheduling problems with consumer jobs, and (iii) variants of the knapsack problem. We will proceed as follows: pick a pair of problems, and prove some approximation preserving reductions in both directions. However, as we will see, the strength of the reductions in the two directions may well be different. The reductions presented are not of mere theoretical interest. Roughly speaking, by reducing a scheduling problem to a knapsack problem, we can use approximation algorithms or heuristics available for solving the knapsack problem as a subroutine for solving the scheduling problem.

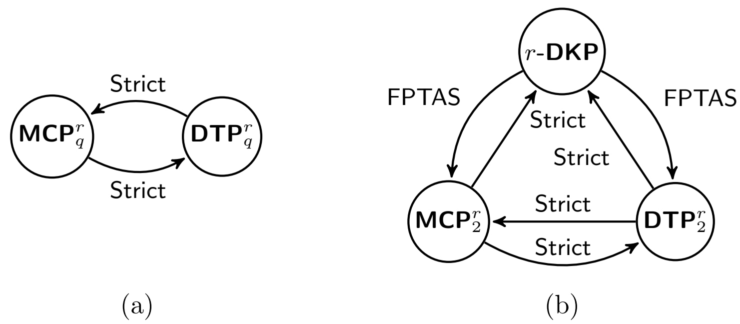

Figure 1: Summary of approximation preserving reductions between scheduling and knapsack problems.

Our findings are summarized in Figure 1 and Table 1. In the figure, a directed arc from problem Π_1 to problem Π_2 labeled by some reduction indicates that Π_1 is reducible to Π_2 by that kind of reduction. In the table we summarize the implications in terms of algorithms of the reductions among the problems. The most important results are: (i) There is a Strict reduction from the problem of minimizing the makespan with consumer jobs, and the scheduling problem with producer jobs and the shifted delivery tardiness objective, and vice versa. This finding allows us to convert approximation algorithms for one type of scheduling problems to the other (part (a) of the figure). (ii) If there are only two supply periods, and a single raw material, then scheduling of consumer jobs to minimize the makespan admits a Strict reduction to the basic knapsack problem (part (b) of the figure), which yields a PTAS as well as an FPTAS for the former problem, i.e., we can use any approximation algorithm devised for the knapsack problem to solve the scheduling problem. (iii) There is no FPTAS for the scheduling problem with consumer jobs and at least two raw materials unless $\mathcal{P} = \mathcal{NP}$, because there is an FPTAS reduction from the multi-dimensional knapsack problem to the scheduling problem (part (b) of the figure), and the multi-dimensional knapsack problem does not admit an FPTAS unless $\mathcal{P} = \mathcal{NP}$ if the number of dimensions is at least two.

The structure of the paper is as follows. We begin with a brief literature review in Section 2, and then we recapitulate basic notions of approximation algorithms and approximation preserving reductions in Section 3. After some preliminaries in Section 4, we establish Strict reduction between MCP_q^r, and DTP_q^r with the shifted delivery tardiness objective (Section 3) in both directions. We proceed with reductions between MCP_2^1 and the knapsack problem in Section 6, and then with reductions between MCP_2^r and the r-Dimensional Knapsack Problem in Section 7 along with consequences in terms of approximability. Finally, we conclude the paper in Section 8.

Problem		PTAS	FPTAS	Source
MCP_2^1	$1\|rm = 1, q = 2\|C_{\max}$	yes	yes	[15], Section 6
MCP_{const}^1	$1\|rm = 1, q = const\|C_{\max}$	yes	?	[15]
MCP_2^r	$1\|rm = r, q = 2\|C_{\max}$	yes	no[a]	Section 7
DTP_2^1	$1\|dm = 1, q = 2\|T_{\max}$	yes	yes	[10], Section 6
DTP_{const}^1	$1\|dm = 1, q = const\|T_{\max}$	yes	?	Section 5
DTP_2^r	$1\|dm = r, q = 2\|T_{\max}$	yes	no[a]	Section 7

[a] if $\mathcal{P} \neq \mathcal{NP}$

Table 1: Approximation schemes for MCP_q^r and DTP_q^r. A questionmark "?" indicates that we are not aware of any definitive answer.

We close this section by the terminology used throughout the paper. For an optimization problem Π, let c_Π denote its cost function, which assigns to every instance I, and feasible solution x to I a scalar value $c_\Pi(I, x)$. Let $R_\Pi(I, x)$ denote the ratio of the optimum value of problem instance I of Π, and the value of some feasible solution x to I. If Π is a maximization problem, then $R_\Pi(I, x) := OPT_\Pi(I)/c_\Pi(I, x)$, while for a minimization problem $R_\Pi(I, x) := c_\Pi(I, x)/OPT_\Pi(I)$. Notice that $R_\Pi(I, x) \geq 1$.

2. Previous work

Scheduling problems with producer jobs only is also known as *scheduling of inventory releasing jobs*, and this model has been recently proposed by Boysen et al. [2]. They studied the problem of minimizing inventory levels while satisfying all the external demands on time (there, the delivery requests have strict deadlines). They proved the NP-hardness of the problem and proposed polynomial algorithms for several variants. Drótos and Kis [10] has introduced the delivery tardiness problem and, among other results, devised an FPTAS for the problem DTP_2^1.

Scheduling of jobs consuming some non-renewable resources (like raw materials, money, energy, etc.) is an old problem class: the original model was described by Carlier [5] and by Carlier and Rinnooy Kan [6] in the early 80's. Since then several authors studied scheduling problems with jobs consuming non-renewable resources (e.g. [26], [28], [24], [14], [3], [12], [4], [15]). In particular, Carlier and Rinnooy Kan [6] defined the problem with precedence constraints, but without machines, and derived polynomial algorithms for various special cases. Carlier [5] showed algorithmic and complexity results. Slowinski [26] studied problems with preemptive jobs on parallel unrelated machines with renewable and non-renewable resources. Toker et al. [28] proved that the problem $1|rm = 1|C_{\max}$ reduces to the 2-machine flow shop problem provided that the resource has a unit supply at each time period. Grigoriev et al. [14] studied problems with one machine and presented some basic complexity results and simple approximation algorithms. Gafarov et al. [12] complemented the findings of Grigoriev et al. by additional complexity results. Neumann and Schwindt [24] studied general project scheduling problems with inventory constraints in a more general setting, where jobs (activities) may consume as well as produce non-renewable resources. In case of a single machine, the problem was proved NP-hard in the strong sense by Kellerer et al. [17], and for minimizing the maximum stock level, the authors proposed three different approximation algorithms with relative error 2, 8/5, 3/2, respectively. Briskorn et al. [3] provided complexity results for several variants, while Briskorn et al. [4] described an exact algorithm for minimizing the weighted sum of the job completion times on a single machine. Györgyi and Kis [15] provided an FPTAS for the problem $1|rm = 1, q = 2|C_{\max}$ and a PTAS for the problem $1|rm = 1, q = const|C_{\max}$.

Knapsack problems are among the most-studied problems in combinatorial optimization. There are many variants and methods of all kinds have been devised over the years to get some solutions, see e.g. the book of Kellerer et al. [20] for an excellent overview. These problems have played an important role in the design of algorithms for scheduling problems, see e.g., [23], [21], [27], [29], [11], [15] to mention but a few examples.

3. Approximation preserving reductions

In this section we recapitulate the basic definitions of approximations schemes, and that of the approximation preserving reductions, and in particular we provide formal definitions of the Strict-, the PTAS-, and FPTAS-reductions. Our discussion follows [8] and [9], see also [1] and [25].

A *Polynomial Time Approximation Scheme* (PTAS) for an optimization problem Π is a family of algorithms $\{A_\varepsilon\}_{\varepsilon>0}$ such that A_ε has polynomial time complexity in the length of any input I for every fixed $\varepsilon > 0$, and always delivers a solution x to I with $R_\Pi(I, x) \leq 1+\varepsilon$. A *Fully Polynomial Time Approximation Scheme* (FPTAS) is a family of algorithms $\{A_\varepsilon\}_{\varepsilon>0}$ with the same properties as a PTAS, plus each A_ε runs in polynomial time in $1/\varepsilon$ as well.

Formally, a *reduction* is a pair of functions f and g, where f maps the instances of problem Π_1 to that of problem Π_2, and g provides a feasible solution for instance I_1 of problem Π_1 from a feasible solution y for the corresponding instance $f(I_1)$ of Π_2. The following diagram illustrates the functions f and g:

$$
\begin{array}{l|ccc}
\text{Problems:} & \Pi_1 & & \Pi_2 \\
\hline
\text{Instances:} & I_1 & \longrightarrow_f & f(I_1) \\
 & & & \big\downarrow \\
\text{Solutions:} & g(I_1, y) & {}_g\longleftarrow & y
\end{array}
$$

(f, g) is a *Strict-reduction* from problem Π_1 to problem Π_2 ($\Pi_1 \leq_{Strict} \Pi_2$) if f and g are computable in polynomial time in the size of their parameters, and for every instance I_1 of Π_1, and for every solution y to $f(I_1)$ we have

$$R_{\Pi_1}(I_1, g(I_1, y)) \leq R_{\Pi_2}(f(I_1), y).$$

A reduction (f, g) is a *PTAS-reduction* from problem Π_1 to problem Π_2 ($\Pi_1 \leq_{PTAS} \Pi_2$) if there exists a function $\alpha(\cdot)$ such that

i) for any instance I_1 of Π_1, and for any $\varepsilon > 0$, $f(I_1, \varepsilon)$ is an instance of Π_2 and it is computable in $t_f(|I_1|, \varepsilon)$ time,

ii) for any solution y of $f(I_1, \varepsilon)$, $g(I_1, y, \varepsilon)$ is a solution to I_1, and it is computable in $t_g(|I_1|, |y|, \varepsilon)$ time,

iii) for every fixed $\varepsilon > 0$, both $t_f(\cdot, \varepsilon)$ and $t_g(\cdot, \cdot, \varepsilon)$ are bounded by a polynomial, and

iv) α maps error parameters for problem Π_1 to that for problem Π_2 such that for every solution y to $f(I_1, \varepsilon)$:

$$R_{\Pi_2}(f(I_1, \varepsilon), y) \leq 1 + \alpha(\varepsilon) \text{ implies } R_{\Pi_1}(I_1, g(I_1, y, \varepsilon)) \leq 1 + \varepsilon. \quad (7)$$

The following statement is from [8].

Lemma 1. *Let Π_1 and Π_2 be optimization problems such that $\Pi_1 \leq_{PTAS} \Pi_2$. If Π_2 admits a PTAS, then there is a PTAS for Π_1 as well.*

The following lemma shows the connection between the Strict-reduction and the PTAS-reduction (for a proof see [9]):

Lemma 2. *Every Strict-reduction is a PTAS-reduction as well.*

Therefore, Lemma 1 remains valid if we replace the PTAS-reduction by Strict-reduction in the statement. Finally, an *FPTAS-reduction* is like a PTAS-reduction with the following modifications:

iii') Both $t_f(\cdot, \varepsilon)$ and $t_g(\cdot, \cdot, \varepsilon)$ must be bounded by a polynomial in $1/\varepsilon$ as well.

iv') α maps *instances* and error parameters for problem Π_1 to error parameters for Π_2 such that for every solution y to $f(I_1, \varepsilon)$:

$$R_{\Pi_2}(f(I_1, \varepsilon), y) \leq 1 + \alpha(I_1, \varepsilon) \text{ implies } R_{\Pi_1}(I_1, g(I_1, y, \varepsilon)) \leq 1 + \varepsilon. \quad (8)$$

That is, (8) replaces (7) in the definition of FPTAS.

v') α can be computed in polynomial time in $|I_1|$ and $1/\varepsilon$.

vi') There exists a two-variable polynomial $poly(\cdot, \cdot)$ such that $1/\alpha(I_1, \varepsilon) \leq poly(|I_1|, 1/\varepsilon)$ for any $\varepsilon > 0$.

Remark 1. *In the above definition, ε may be restricted $0 < \varepsilon \leq c$, where c is a positive constant, since we usually want to choose ε arbitrarily close to 0.*

In [8] the following statement was made:

Lemma 3. *If there is an FPTAS-reduction (f, g) from problem Π_1 to problem Π_2, and if Π_2 admits an FPTAS, then there is an FPTAS for Π_1 as well.*

Observe that an FPTAS-reduction is not a PTAS-reduction in general. To see this, suppose we have a pair of optimization problems Π_1 and Π_2, and there is an FPTAS-reduction from Π_1 to Π_2 with $\alpha(I_1, \varepsilon) := \varepsilon/n$, where n is the number of some objects in I_1, and the n objects in I_1 are mapped to n objects in $f(I_1, \varepsilon)$. Moreover, suppose we have a PTAS for Π_2 of running time $O(n^{1/\omega})$, where ω is the desired error ratio. Now, the running time of the PTAS on instance $f(I_1, \varepsilon)$ with error parameter $\omega := \alpha(I_1, \varepsilon)$ is $O(n^{1/\alpha(I_1,\varepsilon)}) = O(n^{n/\varepsilon})$, which is not polynomial in n. Clearly, a PTAS-reduction is not an FPTAS-reduction in general, since the time complexity of computing f and g is not required to be bounded by a polynomial in $1/\varepsilon$, cf. condition iii) of the PTAS-reduction.

As in the case of PTAS reductions, one can show the following:

Lemma 4. *Every Strict-reduction is an FPTAS-reduction as well.*

The next lemma follows from [8] and [25]:

Lemma 5. *The defined reductions are transitive.*

4. Preliminaries

In this section we overview basic facts about knapsack problems and resource scheduling problems. Consider first the Knapsack Problem KP:

1. We always assume that $w_j \leq b'$, $\forall j = 1, \ldots, n$.
2. There is an FPTAS for KP (see [16] or [19] for faster FPTAS algorithms).
3. There is a 2-approximation algorithm (cf. end of Section 1) for the KP in linear time (see e.g. [20]). There are better approximation algorithms, but these require more time (see [20] for an overview).
4. There is an easily computable upper bound U_{KP} on the optimum value of KP with $OPT_{KP} \leq U_{KP} \leq 2 \cdot OPT_{KP}$. Let $e_j := v_j/w_j$ denote the efficiency of item j. Sort the items by their efficiency in decreasing order (assume that $e_1 \geq \ldots \geq e_n$). Let k be the smallest index such that $w_1 + \cdots + w_k \geq b'$, unless $\sum_{j=1}^{n} w_j < b'$ in which case $k := n$, and let $U_{KP} := v_1 + \cdots + v_k$.
5. We know that $n \cdot OPT_{KP} \geq \sum_{j=1}^{n} v_j$.

As for the r-dimensional Knapsack Problem $r - DKP$:

1. There is a PTAS for $r - DKP$ in [7].
2. $U_{r-DKP} = \sum_{j=1}^{n} v_j$ is an upper bound on OPT_{r-DKP} such that $OPT_{r-DKP} \leq U_{r-DKP} \leq n \cdot OPT_{r-DKP}$.

Finally, some key facts about the Material Consumption Problem. Let S be a schedule for $1|rm = 1, q = 2|C_{\max}$ (MCP_2^1). We say a job j is assigned to the time point u_1 if and only if the total requirement of the jobs that start not later than j in S is at most b_1. Let $P_1(S)$ denote the sum of processing times of these jobs and $P_2(S)$ denote the total processing time of the remaining jobs. Clearly, $P_1(S) + P_2(S) = \overline{P}$, where $\overline{P} := \sum_{j=1}^{n} p_j$.

Observation 1. *Let S^* be an optimal schedule for MCP_2^r. We have*

i) $C^*_{\max} = \max\{P_1(S^*) + P_2(S^*), u_2 + P_2(S^*)\}$.

ii) $C^*_{\max} \geq \overline{P}$ *and* $C^*_{\max} > u_2$.

Proof. Notice that

i) $P_1(S^*) \geq u_2$ implies $C^*_{\max} = P_1(S^*) + P_2(S^*)$ and $P_1(S^*) < u_2$ implies $C^*_{\max} = u_2 + P_1(S^*)$.

ii) $C^*_{\max} \geq \overline{P}$ is obvious from the previous point, and $C^*_{\max} > u_2$ holds because of Assumption 1. □

5. Strict reductions between the Delivery tardiness and the Material consumption problems

In this section we prove that there is a Strict-reduction between DTP_q^r and MCP_q^r in both directions. To illustrate the main idea, we present an example in Figure 2. In the top, there is a schedule for an instance of the DTP_4^1 problem, and in the bottom, a schedule for the MCP_4^1 problem. The rectangles are the jobs, where the horizontal width indicates the processing time, and the vertical height the amount of resource produced (DTP problem), or the material required (MCP problem). The two schedules consist of the same jobs, and the sequence in the bottom is just the reverse of that in the top. The delay in the top indicates the late delivery by job J_{j^*} with respect to due date u_3, whereas in the bottom, the same delay occurs before job J_{j^*} due to waiting for resource supply.

Lemma 6. *Given an instance $I_D = \{n, q, (p_j, a_j)_{j=1}^n, (u_\ell, b_\ell)_{\ell=1}^q\}$ of the Delivery tardiness problem. Define an instance $I_M = \{n, q, (p_j, a_j)_{j=1}^n, (u'_\ell, b'_\ell)_{\ell=1}^q\}$ of the Material consumption problem:*

$$\begin{aligned} u'_\ell &= u_q - u_{q+1-\ell} \\ b'_\ell &= b_{q+1-\ell} \end{aligned} \qquad \ell = 1, \ldots, q.$$

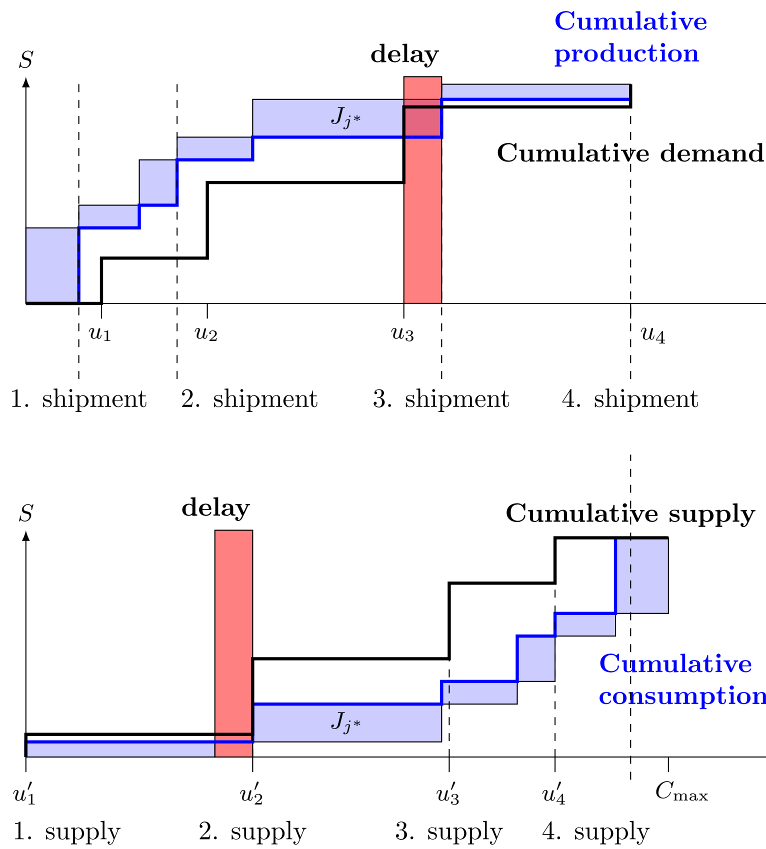

Figure 2: Corresponding schedules for the DTP (top) and MCP (bottom) problems.

Then, if σ is a sequence of jobs giving a maximum delivery tardiness of $T^{\sigma}_{\max}$ for I_D, then scheduling the jobs in reverse σ order gives a schedule of makespan $u_q + T^{\sigma}_{\max}$ for instance I_M.

Proof. Without loss of generality, $\sigma = (J_1, \dots, J_n)$, and then the reverse order of jobs is $\sigma^{-1} = (J_n, J_{n-1}, \dots, J_1)$. For the problem instance I', let $S(\sigma^{-1})$ be the schedule obtained by scheduling the jobs in the order of σ^{-1}, and scheduling each job to start as early as possible while respecting the resource constraints. By contradiction, suppose the makespan $C^{S(\sigma^{-1})}_{\max}$ of schedule $S(\sigma^{-1})$ is larger than $u_q + T^{\sigma}_{\max}$ (notice that u_q is the last due-date of problem instance I of the Delivery tardiness problem). Then by the definition of the makespan, there exist a resource supply date u'_{ℓ^*} and a job index j^* such that

$$C^{S(\sigma^{-1})}_{\max} = u'_{\ell^*} + \sum_{j=1}^{j^*} p_j \tag{9}$$

Take the earliest such ℓ^* and the corresponding index j^*. Since job J_{j^*} is scheduled at the earliest possible time, we also have

$$\sum_{j=j^*+1}^{n} a_j \leq \sum_{\ell=1}^{\ell^*-1} b'_\ell \tag{10}$$

$$\sum_{j=j^*}^{n} a_j > \sum_{\ell=1}^{\ell^*-1} b'_\ell \tag{11}$$

Notice that if $\ell^* = 1$, then since $u'_1 = 0$ by definition, it follows that $j^* = n$ (the makespan is the sum of processing times of all the jobs, since no job may start before time 0), and the right-hand-sides in (10), and (11) are 0. Since $\sum_\ell b_\ell = \sum_j a_j$, (10) and (11) are equivalent to

$$\sum_{j=1}^{j^*} a_j \geq \sum_{\ell=\ell^*}^{q} b'_\ell = \sum_{\ell=\ell^*}^{q} b_{q+1-\ell} = \sum_{\ell=1}^{q-\ell^*+1} b_\ell \tag{12}$$

$$\sum_{j=1}^{j^*-1} a_j < \sum_{\ell=\ell^*}^{q} b'_\ell = \sum_{\ell=1}^{q-\ell^*+1} b_\ell \tag{13}$$

This means that in the instance I of the Delivery tadiness problem, the first $j^* - 1$ jobs are not enough to satisfy the demand of the first $q - \ell^* + 1$ time periods. Since $u'_{\ell^*} = u_q - u_{q-\ell^*+1}$, we have

$$u_q + T^\sigma_{\max} < C^{S(\sigma^{-1})}_{\max} = u'_{\ell^*} + \sum_{j=1}^{j^*} p_j = u_q - u_{q-\ell^*+1} + \sum_{j=1}^{j^*} p_j,$$

where the first inequality follows from our indirect assumption, and the second and third equations from the definition. However, this implies

$$T^\sigma_{\max} < \sum_{j=1}^{j^*} p_j - u_{q-\ell^*+1}.$$

Therefore, schedule σ for instance I_D of the Delivery tardiness problem cannot have maximum tardiness $T^\sigma_{\max}$, a contradiction. □

Lemma 7. *Given an instance $I_M = \{n, q, (p_j, a_j)_{j=1}^n, (u_\ell, b_\ell)_{\ell=1}^q\}$ of the Material consumption problem. Define an instance $I_D = \{n, q, (p_j, a_j)_{j=1}^n, (u'_\ell, b'_\ell)_{\ell=1}^q\}$ of the Delivery tardiness problem:*

$$\begin{aligned} u'_\ell &= u_q - u_{q+1-\ell} \\ b'_\ell &= b_{q+1-\ell} \end{aligned} \qquad \ell = 1, \ldots, q.$$

Then, if S is a schedule with a makespan of $C^S_{\max}$ for I_M, then scheduling the jobs in reverse order (without any delays among them) gives a schedule of maximum tardiness at most $C^S_{\max} - u_q$ for instance I_D.

Proof. Suppose S completes the jobs in the order $\sigma = (J_1, \ldots, J_n)$. The reverse order is $\sigma^{-1} = (J_n, \ldots, J_1)$. Let $S(\sigma^{-1})$ be the schedule corresponding to the reverse order σ^{-1}, i.e., $S_j(\sigma^{-1}) := \sum_{j'=j+1}^{n} p_{j'}$. By contradiction, suppose $T_{\max}(S(\sigma^{-1})) > C^S_{\max} - u_q$. By the definition of $T_{\max}(S(\sigma^{-1}))$, there exist $\ell^* \in \{1, \ldots, q\}$, and some job j^* such that

$$T_{\max}(S(\sigma^{-1})) = \sum_{j=j^*}^{n} p_j - u'_{\ell^*}.$$

Moreover,

$$\sum_{j=j^*}^{n} a_j \geq \sum_{\ell=1}^{\ell^*} b'_\ell \tag{14}$$

$$\sum_{j=j^*+1}^{n} a_j < \sum_{\ell=1}^{\ell^*} b'_\ell \tag{15}$$

Observe that

$$C^S_{\max} - u_q < T_{\max}(S(\sigma^{-1})) = \sum_{j=j^*}^{n} p_j - u'_{\ell^*} = \sum_{j=j^*}^{n} p_j - (u_q - u_{q+1-\ell^*}),$$

which implies

$$C^S_{\max} < u_{q+1-\ell^*} + \sum_{j=j^*}^{n} p_j. \tag{16}$$

In addition (14) and (15) and the assumption $\sum_\ell b_\ell = \sum_j a_j$ imply

$$\sum_{j=1}^{j^*-1} a_j \leq \sum_{\ell=\ell^*+1}^{q} b'_\ell = \sum_{\ell=\ell^*+1}^{q} b_{q-\ell+1} = \sum_{\ell=1}^{q-\ell^*} b_\ell \tag{17}$$

$$\sum_{j=1}^{j^*} a_j > \sum_{\ell=\ell^*+1}^{q} b'_\ell = \sum_{\ell=1}^{q-\ell^*} b_\ell \tag{18}$$

However, (17) and (18) mean that the first j^* jobs in instance I of the Material consumption problem require more resource than that supplied in the first $q - \ell^*$ supply periods. Therefore, the makespan of the schedule is at least $u_{q-\ell^*+1} + \sum_{j=j^*}^{n} p_j$, which is more than the makespan of schedule S by (16), a contradiction. □

Corollary 1. *Let (I_D, I_M) be corresponding instances of the Delivery tardiness and the Material consumption problems. Then the optimum value $T^*_{\max}(I_D)$ of the Delivery tardiness problem equals $C^*_{\max}(I_M) - u_q$, the optimum value of the Material consumption problem minus u_q, where u_q is the last material shipment date in I_M.*

Now we turn to reductions. Since $T_{\max}$ may be 0 in an optimal solution to DTP_q^r, we shift the objective function by a positive constant depending on the problem data: $T_{\max}^s := \max_\ell T_\ell + u_q - u_1$, where u_1 and u_q are the first, and the the last due-date in the DTP_q^r problem instance, respectively. Now we prove the following:

Theorem 1. *There is a Strict-reduction from the Material consumption problem to the Delivery tardiness problem, and vice versa, there is a Strict-reduction from the Delivery tardiness problem to the Material consumption problem.*

PROOF. Firstly, we show that there is a Strict-reduction from MCP_q^r to DTP_q^r. We use the transformation of Lemma 7 to construct function f which maps instances of MCP_q^r to that of DTP_q^r. Clearly, the transformation can be computed in linear time in the size of any instance I_M of MCP_q^r. Let I_M be any instance of MCP_q^r, and let $0 = u_1 < u_2 < \cdots < u_q$ be the dates when some resource is supplied. Then in the corresponding instance $I_D := f(I_M)$ of DTP_q^r, the due-dates are $u'_1 = u_q - u_q = 0$, $u'_2 = u_q - u_{q-1}$, $\ldots$, $u'_q = u_q - u_1 = u_q$. Let σ_D be the order of jobs any solution of instance I_D. The inverse transformation g consists of reversing σ_D. Then, by Lemma 6 we have

$$\begin{aligned}&C_{\max}(S(\sigma_D^{-1})) - u_q + u_q \le T_{\max}(S(\sigma_D)) + u_q = T_{\max}^s(\sigma_D) = (1+\varepsilon)(T_{\max}^s(I_D))^* \\ &= (1+\varepsilon)(T_{\max}(I_D)^* + u_q) = (1+\varepsilon)((C_{\max}^*(I_M) - u_q) + u_q),\end{aligned}$$

where $\varepsilon \ge 0$ is chosen such that $T_{\max}^s(\sigma_D) = (1+\varepsilon)(T_{\max}^s(I_D))^*$, and the second equation follows from $u'_q = u_q$ and $u'_1 = 0$.

Now we prove that there is a Strict-reduction from DTP_q^r to MCP_q^r. We use the transformation of Lemma 6 to construct the function f which maps instances of DTP_q^r to that of MCP_q^r. Let I_D be any instance of DTP_q^r with due-dates $0 \le u_1 < \cdots < u_q$. Then in the corresponding instance $I_M := f(I_D)$ of MCP_q^r, $u'_1 = u_q - u_q = 0$,$\ldots$, $u'_q = u_q - u_1$. Let σ_M be the order of jobs in any solution to I_M. The inverse transformation g reverses the order of jobs in σ_M. We use Lemma 7 to derive

$$\begin{aligned}&T_{max}^s(S(\sigma_M^{-1})) = T_{max}(S(\sigma_M^{-1})) + u_q - u_1 \le C_{\max}(S(\sigma_D)) - u'_q + (u_q - u_1) \\ &= (1+\varepsilon)C_{\max}^*(I_M) = (1+\varepsilon)(T_{\max}^*(I_D) + u'_q) = (1+\varepsilon)(T_{\max}^*(I_D) + u_q - u_1),\end{aligned}$$

where $\varepsilon \ge 0$ is chosen such that $C_{\max}(S(\sigma_D)) = (1+\varepsilon)C_{\max}^*(I_M)$. $\square$

As a consequence, if we manage to get some kind of approximation algorithm from MCP_q^r, then this yields immediately essentially the same algorithm for DTP_q^r with the shifted delivery tardiness objective, and vice versa. Therefore, from now on, we deal with variants of MCP_q^r only.

Corollary 2. *There is a PTAS for DTP_{const}^1.*

PROOF. [15] provided a PTAS for MCP_{const}^1, thus we can apply Lemma 2 and Theorem 1. $\square$

6. Reductions between KP and $\mathbf{MCP_2^1}$

In this section we prove that there is a Strict-reduction from the problem MCP_2^1 to KP and there is an FPTAS-reduction in the opposite direction. Since every Strict-reduction is an FPTAS-reduction as well, we find a new FPTAS for MCP_2^1 with a much better running time than the previously known FPTAS. We start with some preliminary observations.

Lemma 8. *Consider the following two problems :*

Knapsack Problem (KP): *There are n items with profits v_j, item weights w_j $(j = 1, \ldots, n)$, and the knapsack has a capacity of b'.*

Material consumption problem: $1|rm = 1, q = 2|C_{\max}$ *(MCP_2^1) with processing times p_j, resource requirements a_j $(j = 1, \ldots, n)$, and supply dates $0 = u_1 < u_2$, and amount of resource supplied b_1 and b_2 at u_1 and u_2, respectively.*

Suppose $p_j = v_j$, $a_j = w_j$ $(\forall j \in \mathcal{J})$, $b_1 = b'$ and $b_2 = \sum_j a_j - b_1$. Let OPT_{KP} denote the optimum value of KP, and $C^*_{\max}$ that of the Material consumption problem.

i) If $P_1(S^*) < u_2$ for some optimal schedule S^* of the scheduling problem, then $P_1(S') < u_2$, $C^*_{\max} = u_2 + P_2(S')$ and $OPT_{KP} = P_1(S')$ for every optimal schedule S'.

ii) If $P_1(S^*) \geq u_2$ for an optimal schedule S^*, then $C^*_{\max} = P_1(S') + P_2(S') = \overline{P}$ for every optimal schedule S', and $OPT_{KP} \geq u_2$.

Proof. i) Firstly, notice that $P_1(S') = P(S^*)$ for every optimal schedule S', because if there were an optimal schedules S' such that $P_1(S^*) < P_1(S')$, then $P_2(S^*) > P_2(S')$ would follow, and thus $C^*_{\max} = C_{\max}(S^*) = u_2 + P_2(S^*) > \max\{u_2 + P_2(S'), P_1(S') + P_2(S')\} = C_{\max}(S')$, which contradicts the optimality of S^*. Since $P_2(S') = \overline{P} - P_1(S')$, $C^*_{\max} = u_2 + P_2(S')$ follows.

Consider an optimal schedule S^*. Pack the items to the knapsack that correspond to the jobs assigned to u_1 in schedule S^*. Since $b' = b_1$, this is a feasible packing and the total profit is $P_1(S^*)$, therefore $OPT_{KP} \geq P_1(S^*)$.

It remains to prove $OPT_{KP} \leq P_1(S^*)$. Let $\mathcal{K}$ denote the set of the packed items in an optimal solution of KP. Now we build a new schedule S' by scheduling the jobs that correspond to the items in $\mathcal{K}$ in arbitrary order from $t = 0$ without any gaps, and schedule the remaining jobs in arbitrary order from $t = \max\{u_2, p(\mathcal{K})\}$ without any gaps. Since $b_1 = b'$, S' is feasible, hence, $C_{\max}(S') = \max\{u_2 + P_2(S'), P_1(S') + P_2(S')\} \geq u_2 + P_2(S^*) = C_{\max}(S^*)$. Since $P_1(S') + P_2(S') = \overline{P} < u_2 + P_2(S^*)$, as $P_1(S^*) < u_2$ by assumption, we must have $C_{\max}(S') = u_2 + P_2(S')$, and therefore, $OPT_{KP} = p(\mathcal{K}) = P_1(S') \leq P_1(S^*)$.

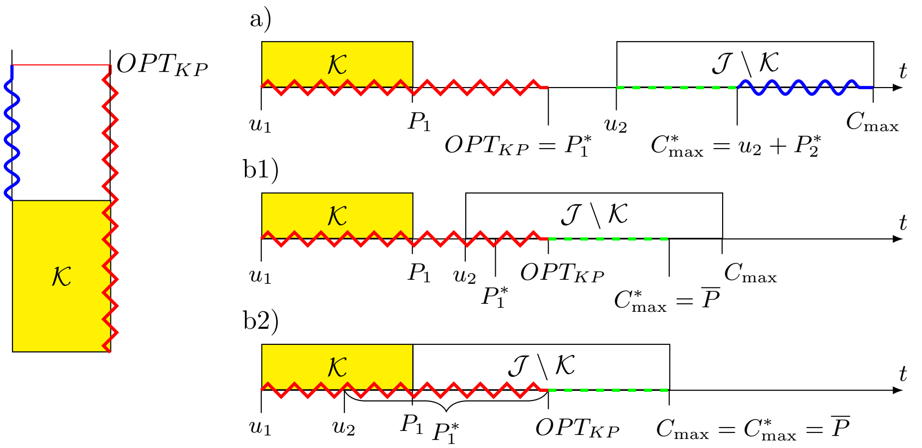

Figure 3: The corresponding solutions of KP and MCP_2^1. On the left: the approximate and optimal solutions of KP (the height indicates the value of a solution). On the right: the approximate and optimal solution of MCP_2^1 in case of a) $P_1^* < u_2$, b1) $P_1^* \geq u_2 > P_1$ and b2) $P_1, P_1^* \geq u_2$. The length of the red zigzag line equals OPT_{KP}, that of the blue wavy line equals $OPT_{KP} - \sum_{j \in \mathcal{K}} p_j$, and the length of the green dashed line is $\overline{P} - OPT_{KP}$.

ii) The first part of the statement is trivial. For the second part consider the schedule S^*. Pack those items into the knapsack that correspond to the jobs assigned to u_1 in schedule S^*. Since S^* is a feasible schedule and $b' = b_1$, this yields a feasible packing for KP of profit $P_1(S^*)$, and thus $OPT_{KP} \geq P_1(S^*)$. Since $P_1(S^*) \geq u_2$ by assumption, we deduce $OPT_{KP} \geq P_1(S^*) \geq u_2$. □

The first main result of this section is a strict reduction from MCP_2^1 to KP. That is, we show that any instance I of MCP_2^1 can be mapped to an instance $f(I)$ of KP in such a way that any solution y of $f(I)$ can be mapped back to a solution $g(I, y)$ of MCP_2^1 with the property that the ratio of the value of the solution $g(I, y)$ and the value of an optimal solution to I is not greater than the ratio of the optimum value of $f(I)$ and the value of the solution y. The idea of the transformation is shown in Figure 3. There is a one-to-one correspondence between the jobs of the scheduling problem, and the items of the corresponding instance of the knapsack problem. Moreover, if $\mathcal{K}$ is the set of items packed into the knapsack in a feasible solution of the KP problem instance, then the corresponding jobs are scheduled consecutively from time 0 on, and the remaining jobs from time $\max\{u_2, \sum_{j \in \mathcal{K}} p_j\}$ on. Let $P_k := P_k(g(I, y))$ and $P_k^* := P_k(S^*)$ for $k = 1, 2$ where S^* is an optimal schedule to I. The three schedules on the right of Figure 3 depend on the relations between P_1, P_1^*, and u_2, and will be elaborated in the proof of the next statement.

Theorem 2. $MCP_2^1 \leq_{Strict} KP$.

Proof. Firstly, we define functions f and g. For a given instance $I = \{n, (p_j, a_j)_{j=1}^n, (u_\ell, b_\ell)_{\ell=1}^2\}$ of MCP_2^1, let $f(I) := \{n, (v_j, w_j)_{j=1}^n, b'\}$ be an instance of KP with $v_j = p_j$, $w_j = a_j$, $j = 1, \ldots, n$, and $b' = b_1$. For a given feasible solution y of instance $f(I)$ of KP, let $\mathcal{K}$ be the set of items that are packed into the knapsack. Define a solution $g(I, y)$ of the Material consumption problem as follows: schedule the jobs that correspond to the items in $\mathcal{K}$ in arbitrary order from time $t = 0$ without any gaps. Define $p(\mathcal{K}) := \sum_{j\in\mathcal{K}} v_j$ which equals $\sum_{j\in\mathcal{K}} p_j$ by the definition of the v_j. Schedule the remaining jobs in arbitrary order after $\max\{u_2, p(\mathcal{K})\}$ without any gaps. Since $b' = b_1$, $g(I, y)$ is a feasible solution of the scheduling problem, and let $C_{\max}$ denote its makespan.

Let y be an approximate solution to $f(I)$. It suffices to prove that for any solution y to the instance $f(I)$ of KP, $R_{MCP_2^1}(I, g(I, y)) \le R_{KP}(f(I), y)$. Let $\varepsilon \ge 0$ be such that $R_{KP}(f(I), y) = 1/(1-\varepsilon)$. Since $R_{KP}(f(I), y) \ge 1$, ε is well defined, and $\varepsilon < 1$. It is enough to show that $R_{MCP_2^1}(I, g(I, y)) \le 1 + \varepsilon$, since $1 + \varepsilon < 1/(1-\varepsilon)$ for any $0 \le \varepsilon < 1$. Let S^* be an optimal schedule, $P_k^* := P_k(S^*)$ for $k = 1, 2$. Let $P_1 := p(\mathcal{K})$, and $P_2 := \overline{P} - P_1$. Using Lemma 8, we distinguish between two cases:

a) $P_1^* < u_2$: in this case $C_{\max}^* = u_2 + P_2^*$, and $OPT_{KP} = P_1^*$ (see Figure 3a) for illustration). By the definition of ε, $P_1 = (1-\varepsilon)OPT_{KP}$. Therefore, $P_2 = \overline{P} - (1-\varepsilon)OPT_{KP}$. Since $P_1^* < u_2$ by assumption, we have $C_{\max} = u_2 + P_2 = u_2 + \overline{P} - (1-\varepsilon)OPT_{KP}$. Since $P_2^* = \overline{P} - P_1^* = \overline{P} - OPT_{KP}$, we have $C_{\max}^* = u_2 + \overline{P} - OPT_{KP}$, hence $C_{\max} = C_{\max}^* + (1-(1-\varepsilon))OPT_{KP} \le (1+\varepsilon)C_{\max}^*$.

b) $P_1^* \ge u_2$: in this case $C_{\max}^* = P_1^* + P_2^* = \overline{P}$, and $OPT_{KP} \ge u_2$, thus $p(\mathcal{K}) \ge (1-\varepsilon)u_2$ (see Figure 3 b1) and b2) for illustration). Then $P_2 \le \overline{P} - (1-\varepsilon)u_2$. Notice that $C_{\max} = \max\{P_1 + P_2; u_2 + P_2\}$ by Observation 1. Since $P_1 + P_2 = \overline{P} = C_{\max}^*$, we only have to prove that $u_2 + P_2 \le (1+\varepsilon)C_{\max}^*$: $u_2 + P_2 \le u_2 + \overline{P} - (1-\varepsilon)u_2 = \overline{P} + (1-(1-\varepsilon))u_2 \le (1+\varepsilon)C_{\max}^*$.

Finally, notice that both of the transformations f and g take linear time and space in the size of I. □

Corollary 3. *There is an FPTAS for MCP_2^1 in $O(n \cdot \min\{\log n, \log(1/\varepsilon)\} + (1/\varepsilon^2)\log(1/\varepsilon) \cdot \min\{n, (1/\varepsilon)\log(1/\varepsilon)\})$ time and in $O(n + 1/\varepsilon^2)$ space.*

Proof. Since every Strict-reduction is an FPTAS-reduction and there is an FPTAS for KP (see e.g. [16]) we can use Lemma 3 to obtain an FPTAS for MCP_2^1. Since the currently best FPTAS for KP requires $O(n \cdot \min\{\log n, \log(1/\varepsilon)\} + (1/\varepsilon^2)\log(1/\varepsilon) \cdot \min\{n, (1/\varepsilon)\log(1/\varepsilon)\})$ time and $O(n + 1/\varepsilon^2)$ space (see [19], [18]), and the transformations f and g take linear time and space, we have proved the complexity results. □

Remark 2. *It has been known that there is an FPTAS for MCP_2^1 (see [15]), but it requires $O(n^7 \cdot 1/\varepsilon^4)$ time and space, therefore the new FPTAS based on the Knapsack Problem is more effective.*

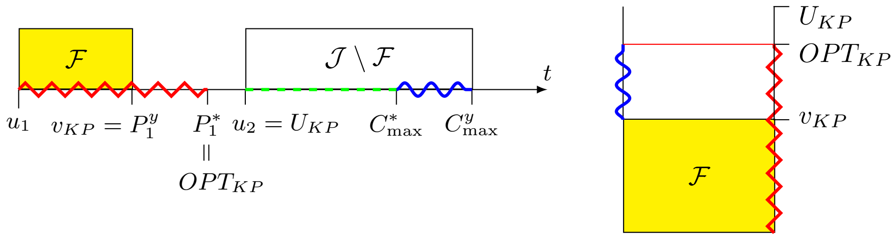

Figure 4: The corresponding solutions of MCP_2^1 (on the left) and KP (on the right, the height of a solution indicates its value). The length of the red zigzag line equals $P_1^* = OPT_{KP}$, that of the blue wavy line equals $OPT_{KP} - v_{KP}$, and the length of the green dashed line is $\overline{P} - OPT_{KP}$.

Corollary 4. *There is an 3/2-approximation algorithm for MCP_2^1 of time complexity $O(n \log n)$.*

Proof. We have shown in the proof of Theorem 2 that if KP admits a $(1/(1-\varepsilon))$-approximation algorithm ($\mathcal{A}$) then MCP_2^1 admits an $(1+\varepsilon)$-approximation algorithm. The complexity of this algorithm is that of $\mathcal{A}$ plus the linear time transformation. Let $\varepsilon := 1/2$ and use the 2-approximation algorithm for KP (see e.g. [20], cf. Section 1). □

Remark 3. *We can create other approximation algorithms for MCP_2^1 if we transform other algorithms originally devised for KP (for an overview of these algorithms see [20]).*

Theorem 3. *$KP \leq_{FPTAS} MCP_2^1$.*

Proof. Let us define functions f and g as follows. For a given instance $I = \{n, (p'_j, w'_j)_{j=1}^n, b'\}$ of KP, let $f(I, \varepsilon) := \{n, (p_j, a_j)_{j=1}^n,\ (u_\ell, b_\ell)_{\ell=1}^2\}$ be an instance of MCP_2^1 with $p_j = p'_j,\ a_j = w'_j,\ j = 1, \ldots, n,\ b_1 = b', b_2 = \sum_{j=1}^n w'_j - b', u_1 = 0, u_2 = U_{KP}$ (where U_{KP} is an upper bound for OPT_{KP} with $OPT_{KP} \leq U_{KP} \leq 2 \cdot OPT_{KP}$, see section 1.1). For a given feasible solution y of instance $f(I, \varepsilon)$ of MCP_2^1, let $\mathcal{F}$ be the set of jobs that are assigned to u_1 in y. Define a solution $g(I, y, \varepsilon)$ of the Knapsack Problem as follows: put the items into the knapsack that correspond to the jobs in $\mathcal{F}$. Let v_{KP} denote the total profit of the items in $\mathcal{F}$. See Figure 4 for illustration.

Since $b_1 = b'$, $g(I, y, \varepsilon)$ is a feasible solution for KP. Notice that the transformation of instance x to $f(I)$ and that of the solution of $f(I, \varepsilon)$ back to a solution of x all take linear time and space in the size of I.

Let $\alpha(I, \varepsilon) := \varepsilon/((1+\varepsilon)(n+1))$, and suppose that y is an $\alpha(I, \varepsilon)$-approximate solution (schedule) to $f(I, \varepsilon)$. We have to show that $g(I, y, \varepsilon)$ is an $(1+\varepsilon)$-approximate solution for KP. Notice that $1/\alpha(I, \varepsilon) = (n+1)(1+\varepsilon)/\varepsilon$ is bounded by a polynomial in $|I|$ and $1/\varepsilon$ for any constant bound on ε (cf. Remark 1 after the definition of the FPTAS-reduction in Section 3). Let $C^y_{\max}$ denote the makespan of the approximate solution y, $P_k^y := P_k(y)$ for $k = 1, 2$, and $\delta := \varepsilon/((1+\varepsilon)(n+1))$. Let S^* be an optimal solution to the scheduling problem of makespan $C^*_{\max}$, and let $P_k^* := P_k(S^*)$ for $k = 1, 2$.

We know that $OPT_{KP} \leq U_{KP}$, thus $P_1^* \leq u_2$, $C^*_{\max} = u_2 + P_2^*$ (see Lemma 8) and $C^y_{\max} = u_2 + P_2^y \leq (1+\delta)C^*_{\max}$. We have $v_{KP} = P_1^y = \overline{P} - P_2^y = \overline{P} + u_2 - C^y_{\max}$. Since $OPT_{KP} = P_1^*$ from Lemma 8, thus $C^*_{\max} = u_2 + \overline{P} - OPT_{KP}$, therefore $v_{KP} \geq \overline{P} + u_2 - (1+\delta)C^*_{\max} = \overline{P} + u_2 - (1+\delta)u_2 - (1+\delta)\overline{P} + (1+\delta)OPT_{KP} = -\delta\overline{P} - \delta u_2 + (1+\delta)OPT_{KP}$. Since $u_2 = U_{KP} \leq 2OPT_{KP}$, $\overline{P} \leq n \cdot OPT_{KP}$ and $\delta > 0$, we deduce that $v_{KP} \geq -\delta n \cdot OPT_{KP} - 2\delta OPT_{KP} + (1+\delta)OPT_{KP} = (1-(n+1)\delta)OPT_{KP} = (1-\varepsilon/(1+\varepsilon))OPT_{KP} = OPT_{KP}/(1+\varepsilon)$. □

Remark 4. *Since the best FPTAS for MCP_2^1 is built on the best FPTAS for KP, this theorem does not have any practical use. However, we can draw an important conclusion from a generalized version of this result for MCP_2^r (see Corollary 8).*

Corollary 5. *$DTP_2^1 \leq_{Strict} KP$ and $KP \leq_{FPTAS} DTP_2^1$.*

Proof. It is a trivial corollary from Lemmas 4, 5 and from Theorems 1, 2 and 3. □

From this we get the following, like we have got Corollaries 3 and 4 from Theorem 2:

Corollary 6. *There is an FPTAS for DTP_2^1 in $O(n \cdot \min\{\log n, \log(1/\varepsilon)\} + (1/\varepsilon^2)\log(1/\varepsilon) \cdot \min\{n, (1/\varepsilon)\log(1/\varepsilon)\})$ time and in $O(n + 1/\varepsilon^2)$ space (it is much better than the previous FPTAS, presented in [10], it requires $O(n^7 \cdot 1/\varepsilon^4)$). There is an 3/2-approximation algorithm for DTP_2^1 of time complexity $O(n \log n)$.*

7. Reductions between r-DKP and MCP$_2^r$

It is easy to generalize the results of the previous sections: there are very similar connections between the problems r-DKP and MCP_2^r. With these results we can prove that there is no FPTAS for the problem MCP_2^r if $r \geq 2$ unless $\mathcal{P} = \mathcal{NP}$. To begin, we generalize Lemma 8 to r-DKP and MCP_2^r:

Lemma 9. *Consider the following two problems :*

r-Dimensional Knapsack Problem (r-DKP): *There are n items with profits v_j, item weights w_{ij} ($i = 1, \ldots, r$; $j = 1, \ldots, n$), and there are capacities of b'_i ($i = 1, \ldots, r$).*

Material Consumption Problem: $1|rm = r, q = 2|C_{\max}$ *(MCP_2^r) with processing times p_j, resource requirements a_{ij} ($i = 1, \ldots, r$; $j = 1, \ldots, n$), and supply dates $0 = u_1 < u_2$, and amount of resource i supplied $b_{1,i}, b_{2,i}$ at u_1 and u_2, respectively.*

Suppose $p_j = v_j$, $a_{ij} = w_{ij}$ *(*$\forall i \in \mathcal{R}$ *and* $\forall j \in \mathcal{J}$*),* $b_{1,i} = b'_i$ *and* $b_{2i} = \sum_j a_{ij} - b_{1,i}$ *(*$\forall i \in \mathcal{R}$*). Let* OPT_{r-DKP} *denote the optimum value of* r*-DKP, and* $C^*_{\max}$ *that of the Material consumption problem.*

i) If $P_1(S^*) < u_2$ *for some optimal schedule* S^* *of the scheduling problem, then* $P_1(S') < u_2$*,* $C^*_{\max} = u_2 + P_2(S')$ *and* $OPT_{r-DKP} = P_1(S')$ *for every optimal schedule* S'*.*

ii) If $P_1(S^*) \geq u_2$ *for an optimal schedule, then* $C^*_{\max} = P_1(S') + P_2(S') = \overline{P}$ *for every optimal schedule* S'*, and* $OPT_{r-DKP} \geq u_2$*.*

Theorem 4. $MCP_2^r \ \leq_{Strict} \ r - DKP$

The proof is identical to that of Theorem 2.

Corollary 7. *For any fixed* r*, there is a PTAS for* MCP_2^r*.*

The corollary follows from a result of [7], which provides a PTAS for r-DKP for any fixed r.

Theorem 5. r*-DKP* $\leq_{FPTAS} MCP_2^r$*.*

The proof is very similar to that of Theorem 3, the crucial difference being that we use Lemma 9 instead of Lemma 8. That is, we let $u_2 = U_{r-DKP}$ in the transformation of an instance of $r - DKP$ to that of MCP_2^r, and we use the bound $U_{r-DKP} \leq n \cdot OPT_{r-DKP}$ in the proof. Remark 5 shows what we can prove exactly:

Remark 5. *For any* $\varepsilon > 0$*, if* MCP_2^r *admits an* $\left(1 + \dfrac{\varepsilon}{(2n-1)(1+\varepsilon)}\right)$*-approximation algorithm, then there is an* $(1+\varepsilon)$*-approximation algorithm for* r*-DKP.*

Corollary 8. *If* $r \geq 2$ *then there is no FPTAS for* MCP_2^r *unless* $\mathcal{P} = \mathcal{NP}$*.*

Proof. If there were an FPTAS for MCP_2^r, then there would exist an FPTAS for $r - DKP$ by Lemma 3 and Theorem 5. However, there is no FPTAS for $2 - DKP$ unless $\mathcal{P} = \mathcal{NP}$ (see [13] or [22]), a contradiction. □

Remark 6. *When* r *is part of the input, no PTAS is known for* $r - DKP$*.*

Corollary 9. $DTP_2^r \leq_{Strict} r - DKP$ *and* $r - DKP \leq_{FPTAS} DTP_2^r$*.*

Proof. Follows from Lemmas 4, 5 and from Theorems 1, 4 and 5. □

Corollary 10. *For any fixed* r*, there is a PTAS for* DTP_2^r*. If* $r \geq 2$ *then there is no FPTAS for* DTP_2^r *unless* $\mathcal{P} = \mathcal{NP}$*.*

8. Conclusions

In this paper we have described approximation preserving reductions among three problem classes, the resource delivery and the material consumption problems, and variants of the knapsack problem. The reductions led to better (faster) algorithms for some special cases of the resource delivery and the material consumption problem, and also to a deeper understanding of the resource delivery and material consumption problems, i.e., the two are essentially the same. We have also shown than neither the material consumption problem, nor the delivery tardiness problem with the shifted tardiness objective admit an FPTAS unless $\mathcal{P} = \mathcal{NP}$.

There remain several open problems: for instance, is there a PTAS for the material consumption problem with a fixed number $r \geq 2$ resources and a fixed number of q time periods? Does there exist an FPTAS for the same problem class with $r = 1$, and $q = 3$? How can we approximate the problem if q or r is not fixed?

Appendix A

$\mathbb{Z}_+$	set of non-negative integers $\{0, 1, 2, \ldots\}$
n	number of jobs
$\mathcal{J}$	set of jobs $\{J_1, \ldots, J_n\}$
p_j	processing time of job J_j
r	number of the resources
$\mathcal{R}$	set of r resources
q	number of due dates (delivery tardiness problem), or number of replenishments (material consumption problem)
u_ℓ	due dates (delivery tardiness problem), or time moments when some resource is supplied (material consumption problem), $0 \leq u_1 < u_2 < \ldots < u_q$
b_ℓ	delivery requirement (delivery tardiness problem), or amount of the resource supplied (material consumption problem) at u_ℓ in case of $rm = 1$
$b_{\ell,i}$	in case of multiple resources, it is like b_ℓ, but for resource $i \in \mathcal{R}$
a_j	resource requirement of job J_j in case of $rm = 1$
$a_{i,j}$	requirement of job J_j from resource i
$\overline{P}$	$\sum_{j=1}^{n} p_j$
$P_1(S)$	the total processing time of the jobs assigned to u_1 in schedule S for $1\|rm, q = 2\|C_{\max}$
$P_2(S)$	$\overline{P} - P_1(S)$
$C^*_{\max}$	the optimal makespan
$T^*_{\max}$	the optimal value of the maximum tardiness

n	number of items
v_j	profit of item j
w_j	weight of item j
b'	capacity of the knapsack
OPT_{KP}	the optimal value of the Knapsack Problem
w_{ij}	weight of item j in the ith constraint of $r-DKP$
b'_i	the capacity of the knapsack of $r-DKP$ in dimension i
OPT_{r-DKP}	the optimal value of $r-DKP$

9. Acknowledgements

The authors are grateful to a referee for constructive comments that helped improve the presentation. This work has been supported by the OTKA grant K112881. The research of Tamás Kis has been supported by the János Bólyai research grant BO/00412/12/3 of the Hungarian Academy of Sciences.

References

[1] G. Ausiello, P. Crescenzi, A. Panconesi, Approximate solution of NP optimization problems, Theoretical Computer Science, 150 (1995) 1–55.

[2] N. Boysen, S. Bock, M. Fliedner, Scheduling of inventory releasing jobs to satisfy time-varying demand: an analysis of complexity, Journal of Scheduling, 16 (2013) 185–198.

[3] D. Briskorn, B.-C. Choi, K. Lee, J. Leung, M. Pinedo, Complexity of single machine scheduling subject to nonnegative inventory constraints, European Journal of Operational Research, 207 (2010) 605–619.

[4] D. Briskorn, F. Jaehn, E. Pesch, Exact algorithms for inventory constrained scheduling on a single machine, Journal of Scheduling, 16 (2013) 105–115.

[5] J. Carlier, Problèmes d'ordonnancements à contraintes de ressources: algorithmes et complexité, Thèse d'état, (1984)

[6] J. Carlier, A. H. G. Rinnooy Kan, Scheduling subject to nonrenewable resource constraints, Operational Research Letters, 1 (1982) 52–55.

[7] A. Caprara, H. Kellerer, U. Pferschy, D. Pisinger, Approximation algorithms for knapsack problems with cardinality constraints, European Journal of Operational Research, 123 (2000) 333–345.

[8] P. Crescenzi, A. Panconesi, Completeness in approximation classes, Information and Computation, 93 (1991) 241–262.

[9] P. Crescenzi, A Short guide to approximation preserving reductions, In: Proceedings of the 12th Annual IEEE Conference on Computational Complexity (CCC '97) 262–273. IEEE Computer Society Press (1997).

[10] M. Drótos, T. Kis, Scheduling of inventory releasing jobs to minimize a regular objective function of delivery times, Journal of Scheduling, 16 (2013) 337–346.

[11] B. Fu, Y. Huo, H. Zhao, Coordinated scheduling of production and delivery with production window and delivery capacity constraints, Theoretical Computer Science, 422 (2012) 39–51.

[12] E. R. Gafarov, A. A. Lazarev, F. Werner, Single machine scheduling problems with financial resource constraints: Some complexity results and properties, Mathematical Social Sciences, 62 (2011) 7–13.

[13] G.V. Gens, E.V. Levner, Computational complexity of approximation algorithms for combinatorial problems, In: Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, 74, pp. 292–300, Springer (1979).

[14] A. Grigoriev, M. Holthuijsen, J. van de Klundert, Basic scheduling problems with raw material constraints, Naval Research of Logistics, 52 (2005) 527–553.

[15] P. Györgyi, T. Kis, Approximation schemes for single machine scheduling with non-renewable resource constraints, Journal of Scheduling, 17 (2014) 135–144.

[16] O.H. Ibarra, C.E. Kim, Fast approximation algorithms for the knapsack and the sum of subset problem, Journal of the ACM, 22 (1975) 463–468.

[17] H. Kellerer, V. Kotov, F. Rendl, G.-J. Woeginger, The Stock Size Problem. Operations Research, 46 (1998) S1–S12.

[18] H. Kellerer, U. Pferschy, Improved dynamic programming in connection with an FPTAS for the knapsack problem, Journal of Combinatorial Optimization, 8 (2004) 5–11.

[19] H. Kellerer, U. Pferschy, A new fully polynomial time approximation scheme for the knapsack problem, Journal of Combinatorial Optimization, 3 (1999) 59–71.

[20] H. Kellerer, U. Pferschy, D. Pisinger, Knapsack Problems, Springer-Verlag (2004).

[21] S.G. Kolliopoulos, G. Steiner, Partially ordered knapsack and applications to scheduling, Discrete Applied Mathematics, 155 (2007) 889–897.

[22] B. Korte, R. Schrader, On the existence of fast approximation schemes, In: A.L. Mangasarian, R.R. Meyer and S.M. Robinson (eds), Nonlinear Programming, 4, pp. 415–437, Academic Press (1981).

[23] E.L. Lawler, Knapsack-like scheduling problems, the Moore-Hodgson algorithm and the 'Tower of Sets' property, Mathematical Computer Modelling, 20 (1994) 91–106.

[24] K. Neumann, C. Schwindt, Project scheduling with inventory constraints, Mathematical Methods of Operations Research, 56 (2002) 513–533.

[25] P. Orponen, H. Mannila, On Approximation Preserving Reductions: Complete Problems and Robust Measures (Revised Version). Technical Report C-1987-28, Department of Computer Science, University of Helsinki, (1990).

[26] R. Slowinski, Preemptive scheduling of independent jobs on parallel machines subject to financial constraints, European Journal of Operational Research, 15 (1984) 366–373.

[27] C.-S. Su, J. C.-H. Pan, T.-S. Hsu, A new heuristic algorithm for the machine scheduling problem with job delivery coordination, Theoretical Computer Science, 410 (2009) 2581–2591.

[28] A. Toker, S. Kondakci, N. Erkip, Scheduling under a non-renewable resource constraint, Journal of the Operational Research Society, 42 (1991) 811–814.

[29] L. Zhang, L. Lu, J. Yuan, Single-machine scheduling under the job rejection constraint, Theoretical Computer Science, 411 (2010) 1877–1882.