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

Péter Györgyi\textsuperscript{a}, Tamás Kis\textsuperscript{b,*}

\textsuperscript{a}Department of Operations Research, Loránd Eötvös University, H1117 Budapest, Pázmány Péter sétány 1/C, Hungary
\textsuperscript{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 $\Pi_1$ to another optimization problem $\Pi_2$ in such a manner

\*Corresponding author

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

Preprint submitted to Theoretical Computer Science September 11, 2014
that if there is a PTAS for \( \Pi_2 \), then this yields a PTAS for \( \Pi_1 \) as well (for formal definitions, see Section 3). So, we can get a positive result for an optimization problem \( \Pi_1 \), i.e., a PTAS, if we can identify another optimization problem \( \Pi_2 \) which admits a PTAS, and if we manage to devise a PTAS reduction from \( \Pi_1 \) to \( \Pi_2 \). On the other hand, if we want to prove that some problem \( \Pi_2 \) does not admit a PTAS unless \( \mathcal{P} = \mathcal{NP} \), it suffices to find another optimization problem \( \Pi_1 \) which does not admit a PTAS unless \( \mathcal{P} = \mathcal{NP} \), and a PTAS reduction from \( \Pi_1 \) to \( \Pi_2 \). 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:

\[
\text{OPT}_{KP} := \max \sum_{j=1}^{n} v_j x_j \quad (1)
\]

\[
\sum_{j=1}^{n} w_j x_j \leq b' \quad (2)
\]

\[
x_j \in \{0, 1\}, \quad j = 1, \ldots, n. \quad (3)
\]

We will use the notation \( \text{OPT}_{KP} \) 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:

\[
\text{OPT}_{r-DKP} := \max \sum_{j=1}^{n} v_j x_j \quad (4)
\]

\[
\sum_{j=1}^{n} w_{ij} x_j \leq b'_i, \quad i = 1, \ldots, r \quad (5)
\]

\[
x_j \in \{0, 1\}, \quad j = 1, \ldots, n. \quad (6)
\]

The optimum value of this problem is denoted by \( \text{OPT}_{r-DKP} \).

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\(^r_q \)) 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 \( J_j, j \in \)
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 \((u_\ell, b_\ell)\) with \( u_\ell \in \mathbb{Z}_+^r \), and \( b_\ell \in \mathbb{Z}_+^r \). The solution of the problem is a sequence \( \sigma \) 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 \((u_\ell, b_\ell)\) is met by \( S \), if the total production of those jobs finishing by \( u_\ell \) is at least \( b_\ell := \sum_{k=1}^r b_k \), i.e., \( \sum_{j : s_j + p_j \leq u_\ell} a_j \geq b_\ell \) (coordinate wise), otherwise it is tardy. Let \( C_\ell(S) \) be the earliest time point \( t \geq 0 \) with \( \sum_{j : s_j + p_j \leq t} a_j \geq b_\ell \). 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 \( 1|r| \text{d} = r|T_{\max}| \), where \( 'd = r' \) 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 \((0 \leq u_1 < u_2)\) when some product is due (denoted by \( 1|r|d = r, q = 2|C_{\max}| \). Since \( T_{\max} \) can be 0 in an optimal solution, we will consider the shifted delivery tardiness objective function defined as \( T^s_{\max} := T_{\max} + \text{const} \), where \( \text{const} \) is a positive constant, depending on the problem data.

In the Material consumption problem \( \text{(MCP)}_{q} \) 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 \( b_j \in \mathbb{Z}_+^r \) 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., \( \sum_{t : u_t \leq S_j} b_t \geq \sum_{j' : s_{j'} \leq S_j} a_{j'} \) (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 \((u_1 = 0 \text{ and } u_2 > 0)\) 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.

Our findings are summarized in Figure 1 and Table 1. In the figure, a directed arc from problem Π₁ to problem Π₂ labeled by some reduction indicates that Π₁ is reducible to Π₂ 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 \( P = \mathcal{NP} \), 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 \( P = \mathcal{NP} \) 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 \( \text{MCP}_q \) and \( \text{DTP}_q \) with the shifted delivery tardiness objective (Section 3) in both direc-

![Figure 1: Summary of approximation preserving reductions between scheduling and knapsack problems.](image-url)
tions. We proceed with reductions between $MCP^1_2$ and the knapsack problem in Section 6, and then with reductions between $MCP^r_2$ and the $r$-Dimensional Knapsack Problem in Section 7 along with consequences in terms of approximability. Finally, we conclude the paper in Section 8.

<table>
<thead>
<tr>
<th>Problem</th>
<th>PTAS</th>
<th>FPTAS</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>$MCP^1_2$</td>
<td>yes</td>
<td>yes</td>
<td>[15], Section 6</td>
</tr>
<tr>
<td>$MCP^1_{\text{const}}$</td>
<td>yes</td>
<td>?</td>
<td>[15]</td>
</tr>
<tr>
<td>$MCP^r_2$</td>
<td>yes</td>
<td>no$^a$</td>
<td>Section 7</td>
</tr>
<tr>
<td>$DTP^1_2$</td>
<td>yes</td>
<td>yes</td>
<td>[10], Section 6</td>
</tr>
<tr>
<td>$DTP^1_{\text{const}}$</td>
<td>yes</td>
<td>?</td>
<td>Section 5</td>
</tr>
<tr>
<td>$DTP^r_2$</td>
<td>yes</td>
<td>no$^a$</td>
<td>Section 7</td>
</tr>
</tbody>
</table>

$^a$if $P \neq NP$

Table 1: Approximation schemes for $MCP^r_q$ and $DTP^r_q$. 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 $\Pi$, let $c_\Pi$ 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 $\Pi$, and the value of some feasible solution $x$ to $I$. If $\Pi$ is a maximization problem, then $R_\Pi(I, x) := \frac{\text{OPT}_\Pi(I)}{c_\Pi(I, x)}$, while for a minimization problem $R_\Pi(I, x) := \frac{c_\Pi(I, x)}{\text{OPT}_\Pi(I)}$. 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 Boyesen 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 $DTP^1_2$.

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 $1|\text{rm}=1|C_{\text{max}}$ 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 $1|\text{rm}=1,q=2|C_{\text{max}}$ and a PTAS for the problem $1|\text{rm}=1,q=\text{const}|C_{\text{max}}$.

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 $\Pi$ is a family of algorithms $\{A_\varepsilon\}_{\varepsilon>0}$ such that $A_\varepsilon$ 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 $A_\varepsilon$ 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 $\Pi_1$ to that of problem $\Pi_2$, and $g$ provides a feasible solution for instance $I_1$ of problem $\Pi_1$ from a feasible solution $y$ for the corresponding instance $f(I_1)$ of $\Pi_2$. The following diagram illustrates the functions $f$ and $g$:
$(f, g)$ is a **Strict-reduction** from problem $\Pi_1$ to problem $\Pi_2$ ($\Pi_1 \leq_{\text{strict}} \Pi_2$) if $f$ and $g$ are computable in polynomial time in the size of their parameters, and for every instance $I_1$ of $\Pi_1$, 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 $\Pi_1$ to problem $\Pi_2$ ($\Pi_1 \leq_{\text{PTAS}} \Pi_2$) if there exists a function $\alpha(\cdot)$ such that

i) for any instance $I_1$ of $\Pi_1$, and for any $\varepsilon > 0$, $f(I_1, \varepsilon)$ is an instance of $\Pi_2$ 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) $\alpha$ maps error parameters for problem $\Pi_1$ to that for problem $\Pi_2$ such that for every solution $y$ to $f(I_1, \varepsilon)$:

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

The following statement is from [8].

**Lemma 1.** Let $\Pi_1$ and $\Pi_2$ be optimization problems such that $\Pi_1 \leq_{\text{PTAS}} \Pi_2$. If $\Pi_2$ admits a PTAS, then there is a PTAS for $\Pi_1$ 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') $\alpha$ maps instances and error parameters for problem $\Pi_1$ to error parameters for $\Pi_2$ such that for every solution $y$ to $f(I_1, \varepsilon)$:

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

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

v') $\alpha$ 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 $\varepsilon > 0$. 

7
Remark 1. In the above definition, $\varepsilon$ may be restricted to $0 < \varepsilon \leq c$, where $c$ is a positive constant, since we usually want to choose $\varepsilon$ arbitrarily close to 0.

In [8] the following statement was made:

Lemma 3. If there is an FPTAS-reduction $(f, g)$ from problem $\Pi_1$ to problem $\Pi_2$, and if $\Pi_2$ admits an FPTAS, then there is an FPTAS for $\Pi_1$ 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 $\Pi_1$ and $\Pi_2$, and there is an FPTAS-reduction from $\Pi_1$ to $\Pi_2$ 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 $\Pi_2$ of running time $O(n^{1/\omega})$, where $\omega$ 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 $1|\text{rm} = 1, q = 2|\text{C}_{\max}$ $(MCP^r_2)$. 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, $P_1(S) + P_2(S) = P$, where $P := \sum_{j=1}^{n} p_j$.

**Observation 1.** Let $S^*$ be an optimal schedule for $MCP^r_2$. We have

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

ii) $C_{\max}^* \geq P$ 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) $C_{\max}^* \geq P$ 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 $DTP^r_q$ and $MCP^r_q$ 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 $DTP^1_4$ problem, and in the bottom, a schedule for the $MCP^1_4$ 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:

$$u_\ell' = u_q - u_{q+1-\ell} \quad \ell = 1, \ldots, q.$$  

$$b_\ell' = b_{q+1-\ell}.$$
Then, if \( \sigma \) is a sequence of jobs giving a maximum delivery tardiness of \( T_{\text{max}}^{\sigma} \) for \( I_{D} \), then scheduling the jobs in reverse \( \sigma \) order gives a schedule of makespan \( u_q + T_{\text{max}}^{\sigma} \) for instance \( I_{M} \).

**Proof.** Without loss of generality, \( \sigma = (J_1, \ldots, J_n) \), and then the reverse order of jobs is \( \sigma^{-1} = (J_n, J_{n-1}, \ldots, J_1) \). For the problem instance \( I' \), let \( S(\sigma^{-1}) \) be the schedule obtained by scheduling the jobs in the order of \( \sigma^{-1} \), and scheduling each job to start as early as possible while respecting the resource constraints.

By contradiction, suppose the makespan \( C_{\text{max}}^{S(\sigma^{-1})} \) of schedule \( S(\sigma^{-1}) \) is larger than \( u_q + T_{\text{max}}^{\sigma} \) (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 \( u'_r \), and a job index \( j^* \) such that

\[
C_{\text{max}}^{S(\sigma^{-1})} = u'_r + \sum_{j=1}^{j^*} p_j \tag{9}
\]
Take the earliest such $\ell^*$ and the corresponding index $j^*$. Since job $J_{j^*}$ is scheduled at the earliest possible time, we also have

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

(10)

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

(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

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

(12)

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

(13)

This means that in the instance $I$ of the Delivery tardiness 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

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

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 $\sigma$ 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:

$$u'_\ell = u_q - u_{q+1-\ell}$$

$$b'_\ell = b_{q+1-\ell}$$

$\ell = 1, \ldots, q.$

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 $I_D$. 

11
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 $\sigma^{-1}$, i.e., $S_j(\sigma^{-1}) := \sum_{j'=j+1}^n p_{j'}$. By contradiction, suppose $T_{\text{max}}(S(\sigma^{-1})) > C_{\text{max}}^S - u_q$. By the definition of $T_{\text{max}}(S(\sigma^{-1}))$, there exist $\ell^* \in \{1, \ldots, q\}$, and some job $j^*$ such that
\[
T_{\text{max}}(S(\sigma^{-1})) = \sum_{j=j^*}^n p_j - u_{\ell^*}.
\]
Moreover,
\[
\sum_{j=j^*}^n a_j \geq \sum_{\ell=1}^{\ell^*} b_{\ell} \quad (14)
\]
\[
\sum_{j=j^*+1}^n a_j < \sum_{\ell=1}^{\ell^*} b_{\ell} \quad (15)
\]
Observe that
\[
C_{\text{max}}^S - u_q < T_{\text{max}}(S(\sigma^{-1})) = \sum_{j=j^*}^n p_j - u_{\ell^*} = \sum_{j=j^*}^n p_j - (u_q - u_{q+1-\ell^*}),
\]
which implies
\[
C_{\text{max}}^S < u_{q+1-\ell^*} + \sum_{j=j^*}^n p_j. \quad (16)
\]
In addition (14) and (15) and the assumption $\sum_{\ell} b_{\ell} = \sum_{j} a_j$ imply
\[
\sum_{j=1}^{j^*-1} a_j \leq \sum_{\ell=\ell^*+1}^q b_{\ell} = \sum_{\ell=\ell^*+1}^q b_{q-\ell+1} = \sum_{\ell=1}^{q-\ell^*} b_{\ell} \quad (17)
\]
\[
\sum_{j=1}^{j^*} a_j > \sum_{\ell=\ell^*+1}^q b_{\ell} = \sum_{\ell=1}^{q-\ell^*} b_{\ell} \quad (18)
\]
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 $T_{\text{max}}^D(I_D)$ of the Delivery tardiness problem equals $C_{\text{max}}^* (I_M) - u_q$, the optimum value of the Material consumption problem minus $u_q$, where $u_q$ is the last material shipment date in $I_M$. 

12
Now we turn to reductions. Since \( T_{\text{max}} \) may be 0 in an optimal solution to \( DTP^r_q \), we shift the objective function by a positive constant depending on the problem data: \( T_{\text{max}} := \max_t T_t + u_q - u_1 \), where \( u_1 \) and \( u_q \) are the first, and the last due-date in the \( DTP^r_q \) problem instance, respectively. Now we prove the following:

**Theorem 1.** There is a PTAS for \( MCP^r_q \).\( \Box \)

Provided a PTAS for \( I \) any solution to \( MCP^r_q \) of \( \alpha \) due-dates \( \sigma \) \( \square \)

We use the transformation of Lemma 6 to construct the function \( f \) which maps instances of \( MCP^r_q \) to that of \( DTP^r_q \). Clearly, the transformation can be computed in linear time in the size of any instance \( I_M \) of \( MCP^r_q \). Let \( I_M \) be any instance of \( MCP^r_q \), 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 \( DTP^r_q \), the due-dates are \( u_q' = u_q - u_1 = 0, u_q'' = u_q - u_{q-1}, \ldots, u_1' = u_1 - u_1 = u_1 \). Let \( \sigma_D \) be the order of jobs any solution of instance \( I_D \). The inverse transformation \( g \) consists of reversing \( \sigma_D \). Then, by Lemma 6 we have

\[
C_{\text{max}}(S(\sigma_D^{-1})) - u_q + u_q \leq T_{\text{max}}(S(\sigma_D)) + u_q = T_{\text{max}}(\sigma_D) = (1 + \varepsilon)(T_{\text{max}}(I_D))^* = (1 + \varepsilon)(T_{\text{max}}(I_D) + u_q) = (1 + \varepsilon)\left( \max_{\sigma_M}(I_M) - u_q \right) + u_q
\]

where \( \varepsilon \geq 0 \) is chosen such that \( T_{\text{max}}(\sigma_D) = (1 + \varepsilon)(T_{\text{max}}(I_D))^* \), and the second equation follows from \( u_q' = u_q \) and \( u_q' = 0 \).

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

\[
T_{\text{max}}(S(\sigma_M^{-1})) = T_{\text{max}}(S(\sigma_M^{-1})) + u_q - u_1 \leq C_{\text{max}}(S(\sigma_D)) - u_q' + (u_q - u_1) = (1 + \varepsilon)\max_{\sigma_M}(I_M) = (1 + \varepsilon)(T_{\text{max}}(I_D) + u_q) = (1 + \varepsilon)(T_{\text{max}}(I_D) + u_q - u_1)
\]

where \( \varepsilon \geq 0 \) is chosen such that \( C_{\text{max}}(S(\sigma_D)) = (1 + \varepsilon)\max_{\sigma_M}(I_M) \). \( \Box \)

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

**Corollary 2.** There is a PTAS for \( DTP^1_{\text{const}} \).

**Proof.** [15] provided a PTAS for \( MCP^1_{\text{const}} \), thus we can apply Lemma 2 and Theorem 1. \( \Box \)
6. Reductions between KP and MCP$_1$

In this section we prove that there is a Strict-reduction from the problem MCP$_2$ 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$_1$ 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|\(rm = 1, q = 2|C_{\text{max}}\) (MCP$_1$) 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 $p_j = v_j$, $a_j = w_j$ (\(\forall j \in J\)), $b_1 = b'$ and $b_2 = \sum_j a_j - b_1$. Let $OPT_{KP}$ denote the optimum value of KP, and $C^*_{\text{max}}$ that of the Material consumption problem.

i) If $P_1(S^*) < u_2$ for some optimal schedule $S^*$ of the scheduling problem, then $P_1(S') < u_2$, $C^*_{\text{max}} = u_2 + P_2(S')$ 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 $C^*_{\text{max}} = P_1(S') + P_2(S') = \overline{P}$ 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 $C^*_{\text{max}} = C_{\text{max}}(S^*) = u_2 + P_2(S^*) > \max\{u_2 + P_2(S'), P_1(S') + P_2(S')\} = C_{\text{max}}(S')$, which contradicts the optimality of $S^*$. Since $P_2(S') = \overline{P} - P_1(S')$, $C^*_{\text{max}} = u_2 + P_2(S')$ 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 $K$ 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 $K$ in arbitrary order from $t = 0$ without any gaps, and schedule the remaining jobs in arbitrary order from $t = \max\{u_2, p(K)\}$ without any gaps. Since $b_1 = b'$, $S'$ is feasible, hence, $C_{\text{max}}(S') = \max\{u_2 + P_2(S'), P_1(S') + P_2(S')\} \geq u_2 + P_2(S^*) = C_{\text{max}}(S^*)$. 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 $C_{\text{max}}(S') = u_2 + P_2(S')$, and therefore, $OPT_{KP} = p(K) = P_1(S') \leq P_1(S^*)$. 

14
The first main result of this section is a strict reduction from MCP$_2$ to KP. That is, we show that any instance $I$ of MCP$_2$ 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$_2$ 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 $K$ 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 $\max\{u_2, \sum_{j \in K} p_j\}$ 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$_2$ \( \leq \text{Strict} \) 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 $MCPC^2$, 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 $K$ 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 $K$ in arbitrary order from time $t = 0$ without any gaps. Define $p(K) := \sum_{j \in K} v_j$ which equals $\sum_{j \in K} p_j$ by the definition of the $v_j$. Schedule the remaining jobs in arbitrary order after max{$u_2, p(K)$} without any gaps. Since $b' = b_1$, $g(I, y)$ is a feasible solution of the scheduling problem, and let $C_{\text{max}}$ 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, $R_{MCPC^2}(I, g(I, y)) \leq R_{KP}(f(I), y)$. Let $\varepsilon \geq 0$ be such that $R_{KP}(f(I), y) = 1/(1-\varepsilon)$. Since $R_{KP}(f(I), y) \geq 1$, $\varepsilon$ is well defined, and $\varepsilon < 1$. It is enough to show that $R_{MCPC^2}(I, g(I, y)) \leq 1 + \varepsilon$, since $1 + \varepsilon < 1/(1 - \varepsilon)$ for any $0 \leq \varepsilon < 1$. Let $S^*$ be an optimal schedule, $P_k^* := P_k(S^*)$ for $k = 1, 2$. Let $P_1 := p(K)$, and $P_2 := P_1 - P_1$. Using Lemma 8, we distinguish between two cases:

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

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

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 $MCPC^2$ 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 $MCPC^2$. 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 $MCPC^2$ (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.
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 ($A$) then MCP$_2^1$ admits an $(1+\varepsilon)$-approximation algorithm. The complexity of this algorithm is that of $A$ 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 $F$ 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 $F$. Let $v_{KP}$ denote the total profit of the items in $F$. 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 $\varepsilon$ (cf. Remark 1 after the definition of the FPTAS-reduction in Section 3). Let $C_{\max}^y$ denote

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 $P - OPT_{KP}$. 
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 $C_{\text{max}}^*$, and let $P_k^* := P_k(S^*)$ for $k = 1, 2$.

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

\[ \square \]

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

**Corollary 5.** $\text{DTP}_2^1 \leq \text{Strict } K^P$ and $K^P \leq \text{FPTAS } \text{DTP}_2^1$.

**Proof.** It is a trivial corollary from Lemmas 4, 5 and from Theorems 1, 2 and 3. \[ \square \]

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

**Corollary 6.** There is an FPTAS for $\text{DTP}_2^1$ 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 $O(n^2 \cdot 1/\varepsilon^4)$). There is a 3/2-approximation algorithm for $\text{DTP}_2^1$ of time complexity $O(n \log n)$.

### 7. Reductions between $r$-$D^P$ and $MCP_2^*$

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

**Lemma 9.** Consider the following two problems:

- $r$-Dimensional Knapsack Problem ($r$-$D^P$): 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: $1|r m = r, q = 2|C_{\text{max}}$ ($MCP_2^*$) with processing times $p_j$, resource requirements $a_{ij}$ ($i = 1, \ldots, r; j = 1, \ldots, n$), and supply dates $b_j = u_1 < u_2$, and amount of resource $i$ supplied $b_{1,i}, b_{2,i}$ at $u_1$ and $u_2$, respectively.
Suppose \( p_j = v_j, a_{ij} = w_{ij} (\forall i \in R \text{ and } \forall j \in J) \), \( b_{1,i} = b'_i \) and \( b_{2,i} = \sum_j a_{ij} - b_{1,i} (\forall i \in R) \). Let \( \text{OPT}_{r-\text{DKP}} \) denote the optimum value of \( r-\text{DKP} \), and \( C^*_{\text{max}} \) that of the Material consumption problem.

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

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

**Theorem 4.** \( \text{MCP}^r_2 \leq_{\text{Strict}} r-\text{DKP} \)

The proof is identical to that of Theorem 2.

**Corollary 7.** For any fixed \( r \), there is a PTAS for \( \text{MCP}^r_2 \).

The corollary follows from a result of [7], which provides a PTAS for \( r-\text{DKP} \) for any fixed \( r \).

**Theorem 5.** \( r-\text{DKP} \leq_{\text{FPTAS}} \text{MCP}^r_2 \).

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 \( u_2 = U_{r-\text{DKP}} \) in the transformation of an instance of \( r-\text{DKP} \) to that of \( \text{MCP}^r_2 \), and we use the bound \( U_{r-\text{DKP}} \leq n \cdot \text{OPT}_{r-\text{DKP}} \) in the proof. Remark 5 shows what we can prove exactly:

**Remark 5.** For any \( \epsilon > 0 \), if \( \text{MCP}^r_2 \) admits an \( \left( 1 + \frac{\epsilon}{(2n - 1)(1 + \epsilon)} \right) \)-approximation algorithm, then there is an \( (1 + \epsilon) \)-approximation algorithm for \( r-\text{DKP} \).

**Corollary 8.** If \( r \geq 2 \) then there is no FPTAS for \( \text{MCP}^r_2 \) unless \( \mathcal{P} = \mathcal{NP} \).

**Proof.** If there were an FPTAS for \( \text{MCP}^r_2 \), then there would exist an FPTAS for \( r-\text{DKP} \) by Lemma 3 and Theorem 5. However, there is no FPTAS for \( 2-\text{DKP} \) unless \( \mathcal{P} = \mathcal{NP} \) (see [13] or [22]), a contradiction. \( \square \)

**Remark 6.** When \( r \) is part of the input, no PTAS is known for \( r-\text{DKP} \).

**Corollary 9.** \( \text{DTP}^r_2 \leq_{\text{Strict}} r-\text{DKP} \) and \( r-\text{DKP} \leq_{\text{FPTAS}} \text{DTP}^r_2 \).

**Proof.** Follows from Lemmas 4, 5 and from Theorems 1, 4 and 5. \( \square \)

**Corollary 10.** For any fixed \( r \), there is a PTAS for \( \text{DTP}^r_2 \). If \( r \geq 2 \) then there is no FPTAS for \( \text{DTP}^r_2 \) 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

<p>| $\mathbb{Z}<em>+$ | 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) |
| $u</em>\ell$ | due dates (delivery tardiness problem), or time moments when some resource is supplied (material consumption problem), $0 \leq u_1 &lt; u_2 &lt; \ldots &lt; u_q$ |
| $b_\ell$ | delivery requirement (delivery tardiness problem), or amount of the resource supplied (material consumption problem) at $u_\ell$ in case of $rm = 1$ |
| $b_{\ell,i}$ | in case of multiple resources, it is like $b_\ell$, 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 $i$ |
| $\mathcal{P}$ | $\sum_{j=1}^{n} p_j$ |
| $P_1(S)$ | the total processing time of the jobs assigned to $u_1$ in schedule $S$ for $1|rm, q = 2|C_{\text{max}}$ |
| $P_2(S)$ | $\mathcal{P} - P_1(S)$ |
| $C_{\text{max}}^<em>$ | the optimal makespan |
| $T_{\text{max}}^</em>$ | the optimal value of the maximum tardiness |</p>
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>number of items</td>
</tr>
<tr>
<td>$v_j$</td>
<td>profit of item $j$</td>
</tr>
<tr>
<td>$w_j$</td>
<td>weight of item $j$</td>
</tr>
<tr>
<td>$b'$</td>
<td>capacity of the knapsack</td>
</tr>
<tr>
<td>$OPT_{KP}$</td>
<td>the optimal value of the Knapsack Problem</td>
</tr>
<tr>
<td>$w_{ij}$</td>
<td>weight of item $j$ in the $i$th constraint of $\text{r-}\text{DKP}$</td>
</tr>
<tr>
<td>$b'_i$</td>
<td>the capacity of the knapsack of $\text{r-}\text{DKP}$ in dimension $i$</td>
</tr>
<tr>
<td>$OPT_{\text{r-}\text{DKP}}$</td>
<td>the optimal value of $\text{r-}\text{DKP}$</td>
</tr>
</tbody>
</table>

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 Bolyai research grant BO/00412/12/3 of the Hungarian Academy of Sciences.

References


